ESTUDIO DE PROCEDIMIENTOS DE AUTENTICACIÓN MULTICANAL BASADOS EN DISPOSITIVOS MÓVILES CON SISTEMA OPERATIVO ANDROID

NICOLÁS MERTANEN CUNÍ

TUTOR: DAVID ARROYO GUARDEÑO, PH. D

MÁSTER EN INGENIERÍA INFORMÁTICA 2013/14

ESCUELA POLITÉCNICA SUPERIOR, UNIVERSIDAD AUTÓNOMA DE MADRID

A mis padres y a mi hermana por estar ahí cuando las cosas iban bien y cuando iban mal.

A Manu, Ricardo, Pelayo, María, Marco, Alejandro, Paula y Gaby por esos momentos de desconexión del trabajo y los estudios.

A David Arroyo por las ideas y las correcciones para llevar este proyecto adelante.

A Blanca por entenderme y darme apoyo todo este tiempo.

Resumen

La evolución de las tecnologías conlleva el aumento de las capacidades de cómputo y procesamiento, ofreciendo, hoy en día, equipos que hace sólo unos años habrían quedado reservados a los grandes centros de investigación. Al igual que se puede dar un gran uso a dichos equipos, su utilización con fines destructivos puede suponer un grave riesgo para el objetivo de dichos ataques. Es por ello que, conforme evoluciona la tecnología, deben evolucionar también los mecanismos de seguridad implantados en los sistemas.

De la misma forma, supone un riesgo igualmente grande la extensión a nuevos usuarios de las herramientas y los mecanismos para llevar a cabo un ataque. La publicación de suites donde se ofrece un compilado de software orientado a la perpetración de ataques no hace sino facilitar su utilización a usuarios menos experimentados.

Uno de los principales objetivos perseguidos por un atacante es eludir los sistemas de control de acceso. El sistema más simple y extendido se corresponde con la asociación de una contraseña para cada uno de los usuarios, pero en el contexto tecnológico actual no resulta seguro hacer uso únicamente de dicho sistema. Actualmente, se recomienda hacer uso de mecanismos de control de acceso estrictos combinados con sistemas de autenticación de varios factores.

En el actual documento se realiza un estudio sobre un sistema al que se quiere dotar de un nivel elevado de seguridad, empleando un sistema multicanal y aplicando un protocolo de autenticación de dos factores para operaciones potencialmente peligrosas.

Abstract

The evolution of technologies in the last decades has implied an increasing of the computational and processing capabilities. This makes possible that the general public can easily access to resources that were restricted to research centers just a few years ago. However, these systems can be used legitimately, but in other cases they are means exploited by malicious users. This is why security mechanisms implemented on these systems must evolve in correspondence to technology improvements.

In the same way, the extension to new users of the tools and mechanisms to carry out an attack represents also a great risk. The publication of suites of software tools endow even less-experienced users with mechanisms to attack computers and information systems.

One of the main goal of attackers is to elude access control schemes. The simplest and more extended access control scheme is defined by assigning secret passwords to each user. In the current technological context is not secure to build up access control just by means of pairs username-secret password. Certainly, strong access control systems should be deployed combining several authentication factors.

In the current document it is performed a study on a system that will be provided with a high level security by using a two-factor authentication protocol to protect critical operations.

Índice de Figuras

Índice de Tablas

Introducción

Este documento realiza un seguimiento de la elaboración del sistema *SecuBank* desde su propuesta y primer estudio hasta su finalización y puesta en funcionamiento. Dicho sistema consiste en una plataforma bancaria donde, para mayor seguridad de las operaciones realizadas, se hace uso de mecanismos adicionales de verificación de la identidad a través de canales adicionales.

I. Motivación

El sistema mencionado se propone con el objetivo de conocer en mayor profundidad los distintos mecanismos de verificación de la identidad actuales, la utilización de diversos canales para el cotejamiento y el modelo de autenticación de dos factores. De la misma forma, se quiere planificar un sistema donde prima la seguridad, de manera que se hace necesario conocer las implicaciones de cada una de las decisiones de diseño o los algoritmos de cifrado seleccionados.

Una vez valoradas las diversas alternativas disponibles, se elige la que presente mayores ventajas de cara a la versión final del sistema, y se procede a llevar a cabo el estudio del sistema y su posterior implementación.

II. Objetivos

El objetivo de este proyecto consiste en el estudio, planificación e implementación de un sistema que permita a sus usuarios la realización de operaciones bancarias seguras, para lo cual se hace uso de unos mecanismos de seguridad que garanticen la identidad de los usuarios. A su vez, dichos mecanismos tienen que ser asequibles al mayor número posible de usuarios, y no suponer una carga para los mismos.

III. Acrónimos

AES	Advanced Encryption Standard
API	Application Programming Interface
APK	Application Package
CCV	Card Code Verification
DES	Data Encryption Standard
GSM	Global System for Mobile Communications
IMEI	International Mobile Equipment Identity
IPC	Inter Process Communication
JSP	JavaServer Pages
MAC	Media Access Control
MD5	Message-Digest Algorithm 5
NFC	Near Field Communication
OS	Operating System
PNG	Portable Network Graphics
PX	Pixel
QR	Quick Response
RAM	Random-Access Memory
RSA	Rivest Shamir Adleman
SDK	Software Development Kit
SMS	Short Message Service
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security
USSD	Unstructured Supplementary Service Data
UTF-8	8-bit Unicode Transformation Format
XML	eXtensible Markup Language
XSS	Cross-Site Scripting

IV. Glosario

- **Array**: Colección de variables almacenadas de manera continua y accedidas mediante un número de índice.
- **Exploit**: Fragmento de software o secuencia de comandos utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado.
- **Fichero Manifest (Android)**: Archivo contenido en un APK. Se usa para definir datos relativos a la extensión, al paquete y a los permisos necesarios para su ejecución.
- **Fuerza Bruta**: Ataque que permite obtener una contraseña de acceso a partir de probar todas las posibles combinaciones hasta que se da con la correcta.
- **Función Hash**: Algoritmo que se aplica sobre una cadena de entrada de longitud variable y genera una nueva cadena de salida de longitud fija, a partir de la cual no se puede deducir la cadena original.
- **HTTP Post**: Solicitud que se realiza sobre el servidor en la que la información está incluida en el propio cuerpo de la solicitud, proporcionando confidencialidad.
- **Intent**: Llamada desde una aplicación en Android a otra actividad, aplicación o proceso.
- **Keylogger**: Software que se encarga de registrar las pulsaciones que se realizan en el teclado, para posteriormente memorizarlas en un fichero o enviarlas a través de internet. También se encuentra hardware destinado a tal fin.
- **Layout**: Determina el esquema de distribución de los elementos dentro del diseño de una aplicación.
- **Licencia Apache**: Licencia de software libre que requiere la conservación del aviso de copyright y el disclaimer, pero no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.
- **Malware**: Software que tiene como objetivo infiltrarse o dañar una computadora o sistema de información.
- **Pixel**: Es la menor unidad que forma parte de una imagen digital. Tiene asociado un valor que determina su color.
- **Salt**: Cadena de bits aleatorios que se usan como entrada junto con la contraseña en el cálculo de la función Hash y almacenamiento de la misma.
- **Servlet**: Clase que amplía las capacidades de una aplicación alojada en un servidor web haciendo uso del lenguaje de programación Java.

- **Smartphone**: Dispositivo móvil con una mayor conectividad que un teléfono móvil convencional y la capacidad de realizar actividades similares a las de un ordenador.
- **String**: Secuencia ordenada de caracteres de longitud arbitraria.
- **Token (software)**: Carácter o cadena de caracteres que tiene o al que se asigna un significado coherente.
- **Token (hardware)**: Dispositivo electrónico que se le da a un usuario autorizado en un sistema para facilitar el proceso de autenticación.
- **Troyano**: Malware con la apariencia de un programa legítimo que le brinda a un atacante acceso remoto al equipo infectado.
- **Virus**: Malware que tiene por objeto alterar el funcionamiento normal del ordenador sin el permiso o el conocimiento del usuario.

Estado de la cuestión

La seguridad es una cualidad de la información que ha importado al hombre a lo largo de toda su existencia, aunque con el paso del tiempo, y la evolución de la sociedad y las tecnologías, el ámbito de aplicación de la misma ha ido viviendo transformaciones constantes.

Desde un primer momento, el concepto de seguridad resulta difícil de definir. En su aplicación más clásica, la idea original de seguridad centraba su radio de acción en garantizar la confianza, mientras que, actualmente, resulta más correcta su aplicación como la responsabilidad de proteger [1].

Ya desde la antigüedad, el hombre se preocupaba de ocultar y proteger sus propiedades de valor. Las primeras llaves y cerraduras se atribuyen a los egipcios hace más de 4000 años [2], al igual que son ampliamente conocidos tanto el cifrado de mensajes por sustitución, cuya invención se atribuye a Julio César y fue ampliamente utilizado en el Imperio Romano [3], como el empleo de esteganografía para ocultar información, extendido también desde la antigüedad.

Partiendo de estas primeras muestras de preocupación por la seguridad y la confidencialidad, tanto las medidas adoptadas como los mecanismos para vulnerar dichas medidas han ido evolucionando conforme lo hacía la tecnología. Desde la creación de jeroglíficos hasta la máquina Enigma. Aunque la verdadera revolución de la seguridad no se ha dado hasta la invención del ordenador, y el descubrimiento de todas las cosas que se pueden llegar a hacer con el mismo.

La aparición de la informática supuso un cambio drástico en la forma de entender la seguridad [4]. Desde un primer momento, cuando cada equipo era accedido por diversos usuarios, éstos se veían obligados a proteger su información haciendo uso de contraseñas. La aparición del

ordenador personal, y especialmente la evolución de Internet, ha dado lugar a toda una nueva serie de servicios ofrecidos al usuario, que deben ser protegidos mediante contraseñas. Actualmente se estima que un usuario normal tiene una media de 40 cuentas en línea [5].

Por otro lado, la evolución de los ataques y la capacidad de los equipos están derivando en una menor seguridad de las contraseñas empleadas por los usuarios. Mientras que una contraseña de 8 caracteres hace 10 años podía considerarse segura, hoy en día se puede llegar a obtener el texto en claro de la misma en cuestión de minutos. Actualmente, la solución a este problema consiste en solicitar al usuario que las contraseñas elegidas cumplan una serie de requisitos: número mínimo de caracteres, uso de mayúsculas y minúsculas, uso de caracteres numéricos y símbolos, además del uso de contraseñas distintas para cada una de las cuentas y servicios de que hace uso, y el cambio de las mismas cada cierto periodo de tiempo [6]. Dado el enorme número de aplicaciones de que hace uso un usuario normal, esto está dando lugar a una situación insostenible, donde nos vemos obligados a recordar y manejar decenas de contraseñas indescifrables e imposibles de recordar, lo cual no hace sino favorecer que el usuario acabe apuntando las mismas en un post-it y pegándolo en el monitor.

El eslabón más débil de la cadena de seguridad de un sistema informático suele ser el usuario final, de manera que la misión de un correcto desarrollador de sistemas es idear una aplicación donde se le ofrezca la mayor cantidad posible de facilidades, en lugar de exigirle una serie de requisitos que, es de sobra conocido, no se pueden ni se van a cumplir.

Además, tal como se ha comentado, a la par que evolucionan las tecnologías, evolucionan los ataques que se pueden llevar a cabo sobre un sistema. Esto no solamente repercute sobre los requisitos que deben cumplir las contraseñas empleadas por los usuarios, sino sobre los sistemas de protección internos de toda la aplicación (cifrado de la información sensible, restricción de acceso a contenido confidencial, tablas de privilegios de usuarios...). Para ganarse la confianza del usuario resulta estrictamente indispensable garantizar la confidencialidad de sus datos.

Actualmente, la única defensa posible contra un ataque realizado por fuerza bruta consiste en el uso de una contraseña de longitud y complejidad suficientemente grande, y siempre teniendo en cuenta que un atacante con una considerable cantidad de tiempo y recursos, siempre acabará obteniendo la contraseña [7].

A este respecto hay que tener en cuenta que el empleo de diccionarios puede llegar a acelerar el proceso considerablemente, ya que se emplean combinaciones de cadenas de caracteres que componen palabras, de manera que todas las contraseñas compuestas por frases (y más por

palabras sueltas), serán descubiertas en un corto periodo de tiempo [8]. De ahí viene la recomendación de las entidades de usar contraseñas con números, letras y símbolos sin un sentido aparente.

Adicionalmente, la aparición de software como *John the Ripper* o *Cain & Abel* está facilitando el acceso del usuario a ataques por fuerza bruta o diccionarios.

Por último, el uso de procedimientos más actuales como los ataques basados en Tablas Arcoíris están reduciendo los tiempos necesarios para perpetrar un ataque a cifras insignificantes. El uso de tablas precompiladas de hashes permite que solamente se tengan que comparar los valores almacenados en la tabla con el correspondiente de la contraseña que se quiere obtener [9]. Dicha comparación se puede llevar a cabo en minutos, en lugar de los días, meses o incluso años que puede llegar a tardar un ataque por fuerza bruta. La forma más extendida para evitar este ataque es la utilización de un Salt o valor aleatorio generado en el momento de guardado de la contraseña, y concatenado a ésta, para evitar así que un mismo texto genere siempre el mismo hash [10].

De la misma forma, los servidores se tienen que mantener actualizados para evitar las vulnerabilidades que se van descubriendo, se deben establecer unas políticas de permisos y privilegios sólidas para evitar el acceso a contenido no autorizado, y se debe disponer de una planificación de seguridad donde se detalle el procedimiento a seguir para cada una de las posibles incidencias que puedan darse [11].

A su vez, la interfaz web con la que interactúa el usuario también debe estar correctamente implementada para evitar posibles filtraciones de información. Ataques como la Inyección SQL o Cross Site Scripting están ampliamente difundidos, de manera que existen múltiples soluciones para evitar sus efectos [12], pero una página web donde no se haya previsto el ataque haciendo uso de dichas vulnerabilidades puede poner en bandeja toda la información del sistema a un atacante.

Por otro lado, desarrollar un sistema compatible con dispositivos móviles implica cambiar el prisma desde el que se orienta el estudio de la seguridad. La larga trayectoria de ataques, vulnerabilidades, virus y troyanos que se han dado en PC a lo largo de la historia han derivado en que el usuario común tenga cierta reticencia y desconfianza a todo lo que rodea la utilización de información personal. Pero, por alguna razón, dicha idea no es trasladada a los dispositivos móviles. El usuario normal se siente confiado haciendo uso del Smartphone. En la mayoría de casos, no son apenas conscientes de la enorme cantidad de ataques que se pueden llevar a cabo a través del terminal móvil, y de la cantidad de información confidencial que se puede filtrar por

no hacer caso de las políticas de acceso y uso de los recursos del dispositivo que solicitan las aplicaciones.

Como dicho usuario, en la mayoría de los casos, no presta tanta atención a la seguridad en su dispositivo móvil [13], resulta especialmente importante implementar las aplicaciones orientadas a este sistema garantizando una completa confidencialidad y privacidad de los datos y las comunicaciones establecidas.

1.1 Dispositivos móviles

El uso de dispositivos móviles en la actualidad se encuentra enormemente extendido. Desde los primeros terminales, diseñados exclusivamente para la realización de llamadas desde cualquier sitio, el teléfono móvil ha vivido una evolución que ha integrado el uso de distintas tecnologías en un mismo dispositivo.

La idea base sobre la que se sustenta su utilización es la posibilidad de realizar y recibir llamadas de teléfono desde cualquier lugar haciendo uso de una red inalámbrica de radio. Dicha red está compuesta por una serie de centrales y repetidores, cada uno de los cuales define una celda o área de cobertura. La unión de todas las celdas conforma la red de comunicaciones, que permite interconectar cualquier dispositivo que se encuentre dentro del área cubierta por dicha red [14] [15].

Figura 1-1: División en celdas del terreno

Esta tecnología empezó a ser explotada en los años 80, para posteriormente, en los 90, desarrollar toda una serie de añadidos sobre la idea original, mejorando así las capacidades de los primeros móviles. Uno de los primeros añadidos fue el envío de mensajes de texto haciendo

uso de la misma red de comunicaciones, aunque la pantalla a color y la cámara no tardaron en llegar [16] [17], y a pesar de que actualmente integra una serie de servicios que anteriormente se asociaban con dispositivos diferenciados, como el GPS, las cámaras de alta definición, la pantalla táctil o el reproductor de música, el verdadero concepto que ha permitido la evolución de los dispositivos móviles llevándolos al punto en que se encuentran hoy en día es la integración de internet de alta velocidad [18].

El uso de internet y el consiguiente nacimiento del Smartphone han supuesto una de las mayores revoluciones tecnológicas y que, en mayor medida, han afectado a la vida diaria de las personas. No es sólo el uso de los dispositivos móviles para mantener un contacto constante con otras personas a través de aplicaciones de mensajería instantánea o redes sociales. El Smartphone permite, prácticamente, la sustitución absoluta del ordenador personal para la mayor parte de servicios que ofrece, especialmente para el usuario común. Esto implica disponer de todos los servicios necesarios para comunicarse en un dispositivo que se lleva encima en todo momento, en lugar de depender de tener acceso a un equipo fijo con conexión a internet.

Figura 1-2: Evolución del Motorola Dynatac al Nexus 5

El origen del Smartphone o teléfono inteligente no está claramente delimitado, aunque parece ser que los primeros dispositivos móviles considerados como Smartphones permitían al usuario tanto la consulta de su correo electrónico como la instalación adicional de programas sobre el dispositivo, aumentando así la capacidad de procesamiento de datos o de entretenimiento [19]. Estas son las características principales que empezaron a anunciar, en 2007, los primeros terminales que terminarían evolucionando en el móvil de hoy en día, ofreciéndose inicialmente

como un dispositivo orientado a empresas y usuarios con altas necesidades de conexión, generalmente por motivos laborales.

1.1.1 Bring your own device (BYOD)

Tal como se ha visto, la revolución iniciada con el Smartphone está afectando en todos los sentidos posibles a la vida de las personas. Dicha revolución, al igual que en el ámbito personal, se está acentuando en el sector empresarial, donde cada vez es más común que los propios empleados hagan uso de sus propios dispositivos para acceder a los recursos de la empresa [20]. La aplicación más común de este concepto es la configuración y uso del correo corporativo en el teléfono personal del usuario, aunque esta ideología se está extendiendo conforme aumenta el número de dispositivos que los usuarios pueden llevar al trabajo. El uso de tablets y ordenadores portátiles personales para acceder a los recursos, aplicaciones y datos de la empresa está a la orden del día, lo cual implica la instalación de las aplicaciones y servicios necesarios sobre un dispositivo sobre el que la empresa no tiene control alguno. Además de la desconfianza que esto pueda suponer a la empresa, esta práctica conlleva una serie de implicaciones de seguridad que afectan de manera directa a la confidencialidad de la información [21] [22]. Ya no es sólo la posible pérdida de un dispositivo que permite acceso a contenido confidencial, es la posibilidad de infección del mismo dispositivo por algún virus o troyano derivado del uso personal del mismo, o su utilización en entornos no seguros, como redes abiertas, donde se puede dar la monitorización o captura no autorizada de tráfico.

Por otro lado, el uso de dispositivos personales aporta una serie de ventajas a la empresa. El número de dispositivos con que debe abastecer a los empleados cada vez es menor, y se dispone de un mayor control de dichos empleados [23], que pueden llegar a realizar tareas relacionadas con el trabajo incluso fuera de su horario laboral. A su vez, actualmente se cree que el uso de dispositivos personales ayuda a los empleados a ser más productivos [23] y, al tratarse de una nueva tendencia, da una imagen moderna y flexible de la empresa al exterior. Es por esto, que las empresas están intentando delimitar el uso de esta serie de dispositivos, permitiendo su utilización pero intentando controlar las condiciones en las que se da su uso.

Algunas de las medidas que se recomienda tomar para garantizar la seguridad de la información de la empresa al permitir el concepto BYOD son las siguientes [24]:

- El acceso a recursos de la empresa debe estar protegido mediante credenciales que permitan conocer quién accede y qué información manipula.
- La gestión se debe basar en roles, de manera que cada usuario solamente acceda a los recursos que realmente necesite.

- Se debe hacer un análisis de qué dispositivos son los que resultan más adecuados para manejar la información de la empresa.
- Se debe garantizar que todos los dispositivos dispongan de soluciones de seguridad adecuadas para evitar amenazas sobre los datos de la empresa.
- El acceso a través de conexiones WiFi debe ser configurado haciendo uso de protocolos de cifrado, para garantizar la seguridad de las comunicaciones.
- Se debe hacer uso de tecnologías como VPN para acceso remoto, que garantiza la protección de la información que se manipula.
- Realizar análisis de riesgos y vulnerabilidades de manera frecuente para conocer el estado de la empresa.
- Educar a los usuarios para asegurarse que conocen tanto los riesgos como los cuidados necesarios para garantizar la seguridad.

1.1.2 Mobile Device Management (MDM)

La popularidad y el crecimiento de la política BYOD está derivando en una dificultad cada vez mayor para llevar un seguimiento exhaustivo de todos los dispositivos que cada uno de los empleados utiliza en la empresa. El objetivo de dicho seguimiento es el de comprobar que se cumplen cada uno de los requisitos anteriormente vistos, para lo cual se está obligando a la utilización de agentes adicionales para garantizar que dichos dispositivos cumplen la normativa establecida por la empresa. Dichos agentes suelen estar conformados por un software de monitorización bajo el nombre de Mobile Device Management, o administración de dispositivos móviles.

El MDM es un software que, tal como su nombre indica, permite la gestión de toda una serie de dispositivos móviles, ofreciendo la monitorización de sus actividades, rastreo y localización de dispositivos, instalación automática de aplicaciones, sincronización de información contenida en el mismo e, incluso, la toma de control del dispositivo de manera remota, además de asegurar la seguridad de éste [25].

Al asegurarse tanto el contenido del dispositivo como las conexiones realizadas a través de éste, la empresa puede permitir la conexión del mismo a la red interna, sin poner en riesgo la información contenida en dicha red.

Generalmente, la arquitectura de dichas aplicaciones se compone de un cliente y un servidor. El cliente se instala sobre cada uno de los dispositivos a controlar, mientras que el servidor solicita los comandos a realizar a los clientes [26]. La información solicitada a los mismos es enviada al

servidor haciendo uso, generalmente, de la red de conexión del propio dispositivo (3G, GPRS, WiFi), para posteriormente almacenarla en la base de datos del propio servidor.

El control de los dispositivos se realiza de manera centralizada, de forma que el administrador hace uso de una consola de administración para configurar o actualizar, de modo remoto, un dispositivo individual, grupo de dispositivos o conjunto de grupos. Generalmente, las opciones de control que tiene a su disposición son las siguientes [27] [28]:

- Instalación masiva de aplicaciones y ejecución de actualizaciones, de manera remota y controlando los parámetros de conexión.
- Aplicación de políticas de restricción de aplicaciones para evitar la instalación de aplicaciones determinadas en los dispositivos.
- Rastreo y localización de dispositivos en un momento concreto haciendo uso del GPS, así como seguimiento de la ruta trazada por un dispositivo.
- Sincronización de los archivos contenidos en el dispositivo con el servidor para llevar a cabo su monitorización y evitar su pérdida.
- Bloqueo de funciones propias del dispositivo, como el uso de la cámara, micrófono, acceso a la configuración del dispositivo o conexión del mismo a través de USB.
- Control del consumo realizado por el dispositivo, que incluye la monitorización de llamadas y uso de internet, y permite la configuración de alertas.
- Eliminación de información del dispositivo de manera remota, utilizado en caso de pérdida o extravío del mismo para evitar la filtración de información confidencial.
- Configuración de una contraseña de manera remota para permitir desbloqueo del dispositivo.

1.2 Sistema Operativo Android

Android es un sistema operativo originalmente orientado a Smartphones, aunque actualmente se encuentran versiones para tablets, netbooks o consolas. Está basado en el kernel de Linux, y el funcionamiento interno del sistema está codificado en C y C++, aunque las aplicaciones desarrolladas para este sistema se implementan en Java. Dispone de una máquina virtual Dalvik, que permite compilación en tiempo de ejecución [29].

Originalmente, Android fue desarrollado por la empresa Android Inc., que posteriormente fue comprada por Google en 2005. Ésta respaldó el proyecto, que fue presentado oficialmente en noviembre de 2007, y liberó la mayor parte del código bajo la licencia Apache. El primer

Smartphone que montaba sistema operativo Android de fábrica fue el HTC Dream, que salió a la venta en octubre de 2008 [30] [31].

A finales del año 2011, Android ya superaba el 50% de la cuota mundial de mercado, situándose por encima del doble de su inmediato perseguidor, iOS. Esto supone uno de los mayores crecimientos de mercado vistos hasta la fecha [31].

Actualmente, el sistema operativo Android acapara la mayor cuota del mercado de Smartphones de acuerdo con la *International Data Corporation* (IDC). Al finalizar 2013, Android disponía del 78.1% del total de ventas, seguido de iOS con un 17.6% [32].

Parte de su éxito se debe a las licencias de código abierto bajo las que se ampara el sistema, que permiten a cualquier usuario no sólo desarrollar nuevas aplicaciones para los dispositivos Android, sino modificar a su antojo el núcleo del sistema operativo. Aunque el mayor atractivo radica, efectivamente, en el desarrollo de aplicaciones.

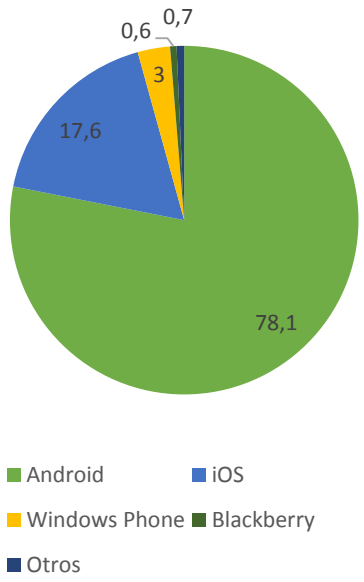


Figura 1-3: Cuota de Mercado de Smartphones

Dichas aplicaciones se desarrollan en el lenguaje de programación Java, haciendo uso del Android Software Development Kit (SDK). Existen versiones de los principales entornos de desarrollo orientados a la creación de aplicaciones para Android, que ya vienen con el Kit de desarrollo preconfigurado. Una vez desarrolladas, se empaquetan en el contenedor APK, que permite su fácil manejo, instalación y distribución en los principales mercados de aplicaciones.

A su vez, Android ofrece Google Play, actualmente la mayor tienda de aplicaciones para Smartphone, que en Junio de 2014 sumaba un total de 1.271.054 aplicaciones disponibles para su descarga (según el apartado *Stats* de la web *AppBrain*).

El precio para conseguir una licencia de desarrollador de aplicaciones Android y poder publicar en la tienda de aplicaciones es de $25, lo cual es una cifra asequible para cualquiera e implica una serie de ventajas y desventajas.

Por un lado, se ofrece la posibilidad de lanzarse al mercado a pequeñas empresas y desarrolladores individuales que desean ofrecer sus servicios y aplicaciones, y que de otra forma podrían no tener los recursos necesarios para hacerlo. Por otro lado, se da la opción de que el mercado se llene de una serie de aplicaciones de una calidad ínfima o, incluso, que puedan

resultar dañinas para los usuarios que se las descarguen. Obviamente, toda aplicación detectada como peligrosa implica el cierre automático de la cuenta del desarrollador correspondiente, pero dado el bajo precio de apertura de una nueva cuenta, esto no debe suponer un problema para los desarrolladores de aplicaciones maliciosas.

1.3 Seguridad en dispositivos Android

A diferencia de Apple, que aprueba de manera manual cada una de las aplicaciones que se ofrecen en la App Store, Google simplemente realiza un escaneo de las aplicaciones en busca de malware antes de publicarlas en Google Play. Esto les permite detectar todas las aplicaciones que hacen uso de código malicioso conocido, pero no reporta ningún aviso cuando se trata de nuevas funcionalidades, o cuando se hace uso de servicios, como el envío de SMS, para subscribir al usuario a servicios Premium de manera oculta [33].

Por otro lado, el sistema operativo Android permite la instalación de aplicaciones no verificadas, descargadas de internet, a través de paquetes APK. Su ejecución en el dispositivo puede suponer un riesgo aún mayor, ya que dichas aplicaciones no han superado ninguna clase de verificación.

Figura 1-4: Verify Apps sobre Android 4.4 KitKat

Android ha añadido la característica Verify Apps, que viene instalada de manera nativa en el terminal a partir de la versión 4.2 Jelly Bean, y es compatible para instalar en todos los dispositivos que dispongan de una versión de Android 2.3 o superior. Se trata de una aplicación que analiza las instalaciones que se realizan sobre el Smartphone e informa al usuario cuando se detecte que una app puede suponer un riesgo para el dispositivo. Dicha herramienta, además, realiza un escaneo constante del teléfono, para detectar posibles cambios en las aplicaciones que no se dieran en el momento de su instalación, evitando así que con alguna actualización se pueda añadir funcionalidad dañina a aplicaciones existentes [34].

Una característica poco explotada por los usuarios, pero que aporta una gran cantidad de información a la hora de evitar un posible ataque, consiste en la identificación de los permisos que requiere una aplicación antes de llevar a cabo su instalación. Cuando se está desarrollando

una aplicación para Android, todo acceso a recursos externos a la aplicación se debe detallar en el fichero Manifest de la aplicación para que el sistema operativo permita el acceso a dichos recursos. Luego estos quedan reflejados en el momento de instalación de la aplicación. Si un usuario cuidadoso analiza los permisos que solicita una aplicación antes de instalarla, e identifica que algunos de los servicios no cuadran con el objetivo de la aplicación, puede evitar, de manera rápida, la instalación de un software que acceda a recursos, información o funcionalidad que no le corresponde.

1.3.1 Ataques a dispositivos Android

La gran cantidad de información personal y confidencial que se almacena en un Smartphone lo convierte en un dispositivo sumamente atractivo para atacantes, lo cual ha derivado en una creciente cantidad de ataques y malware específicamente orientado a este tipo de dispositivos, que suelen tener sus bases en ataques anteriormente conocidos y explotados en equipos de sobremesa.

Los posibles ataques se pueden clasificar, según su objetivo, de la siguiente manera [35]:

- **Robo o substracción de información**: El atacante busca obtener información personal o datos concretos almacenados en el dispositivo. A su vez, este ataque se puede subdividir en función del tipo de información que se desea conseguir:
    - **Nombres de usuario y contraseñas**: Permiten el acceso a cuentas personales del usuario, y la consiguiente suplantación de su identidad. No sólo permite la obtención de información personal del usuario, sino que facilita la difusión no autorizada de webs de contenido malicioso haciendo uso de la cuenta del usuario afectado, o la obtención de información de carácter más sensible, como datos bancarios.
    - **Datos de formularios**: Otro de los objetivos atractivos para los atacantes. Al almacenar la información de los formularios en la memoria del dispositivo para evitar su escritura de forma repetida, ésta queda accesible para un posible atacante. La introducción de datos bancarios para la realización de compras online puede suponer un grave riesgo.
    - **Documentos privados**: Al igual que la información anterior, un atacante que tenga control sobre un dispositivo puede acceder a fotos, documentos, SMS y demás archivos almacenados en el mismo. De esta forma, pueden darse casos de espionaje industrial, en el caso de dispositivos empresariales, o secuestro de información en el ámbito personal.

- **Beneficio lucrativo**: El atacante busca obtener un beneficio monetario de manera directa a través del ataque realizado sobre el dispositivo. Aunque, generalmente, todos los ataques terminan por tener un objetivo lucrativo, aquí se detallan los ataques donde el objetivo estricto del ataque es la obtención de dinero:
    - **Mensajería Premium**: A diferencia de los ataques anteriores, el atacante no obtiene acceso sobre el dispositivo. Este ataque consiste en la instalación de un malware, generalmente camuflado bajo el aspecto de un software confiable, que realiza el envío de mensajes instantáneos a números Premium de manera automática e indetectable para el usuario, generando un beneficio económico al atacante.
    - **Secuestro del dispositivo**: Consiste en el bloqueo del dispositivo completo, de parte de su funcionalidad, o el cifrado de la información contenida en el mismo. Una vez realizado el ataque, se solicita un código para restaurar el estado anterior del dispositivo, para lo cual la víctima se debe poner en contacto con el atacante.
- **Satisfacción personal**: Por último, un atacante puede buscar, simple y llanamente, la satisfacción de haber realizado un ataque de forma satisfactoria. Aunque esta mentalidad estaba más extendida en los años 80, con la informática en una fase más temprana de su madurez, siguen dándose casos donde el atacante busca simplemente demostrar al mundo lo que puede hacer:
    - **Botnets**: Una botnet es una red de dispositivos infectados, también conocidos como terminales zombie. Dicha red está controlada por un nodo central, que permite la ejecución de comandos de manera remota sobre los dispositivos. Las botnets generan al atacante una gran capacidad de cómputo, que generalmente se utiliza para la rotura de contraseñas, el envío de spam o la realización de ataques de denegación de servicio.
    - **Demostración de capacidad**: Se da cuando se lleva a cabo un ataque que destaca por su complejidad. En este caso, el atacante únicamente busca el reconocimiento de la comunidad. Generalmente, este tipo de ataques sientan las bases para que posteriormente se desarrollen otros más elaborados.

Todo este tipo de ataques se difunden a través de medios confiables por el usuario, donde el atacante suplanta la identidad de una fuente conocida por la víctima, o simplemente intenta aparentar ser un recurso legal y espera a ver si el usuario cae en la trampa. Algunos de los medios más utilizados son los siguientes:

- **Redes sociales**: Generalmente, se ofrece al usuario algún recurso o funcionalidad adicional no disponible en la propia red social, como un juego o el clásico "Descubre quién te tiene bloqueado". Cuando el usuario accede al enlace, se le solicitan sus datos de acceso a la red social para poder ofrecerle dicho recurso. En el momento en que rellena dichos datos, la información pasa directamente a manos del atacante, que podrá suplantar la identidad del usuario, acceder a sus datos personales, perfil, correo electrónico y demás información.
- **Correo electrónico**: Otro medio clásico de infección. El atacante envía un mensaje a la víctima o víctimas (generalmente disponen de enormes listas de direcciones de correo) instándole a que abra o ejecute un fichero que contiene un malware camuflado. En el instante en que dicho programa es ejecutado, se abre una puerta de acceso al dispositivo para el atacante.
- **Tiendas de aplicaciones**: Tal como se ha comentado, actualmente se dispone de tiendas de aplicaciones online para Smartphone, donde los usuarios tienen la capacidad de subir las aplicaciones desarrolladas por ellos para que otros se las descarguen. A estas alturas se han dado muchos casos en los que las aplicaciones ofrecidas presentaban algún tipo de infección o malware, ya que el análisis realizado sobre dichas apps no suele ser muy exhaustivo. Se debe prestar especial atención en estos casos, ya que que la aplicación se ofrezca en un medio legal no quiere decir que sea legítima. Huelga decir que las aplicaciones ofrecidas en mercados alternativos o los conocidos como "mercados negros" presentan un índice de riesgo de infección infinitamente mayor.
- **Infección por red**: Al encontrarse continuamente conectados a internet o diversas redes, los Smartphones son susceptibles de sufrir ataques a través de la conexión:
    - **WiFi**: Un dispositivo malicioso conectado a una red WiFi puede realizar un barrido de otros dispositivos conectados a dicha red con la intención de infectarlos. Además, si no se trata de una red segura, o directamente es una red abierta, un atacante puede estar realizando una monitorización del tráfico web no cifrado.
    - **Bluetooth**: De la misma forma, los ataques realizados por bluetooth en dispositivos móviles son aún anteriores. Un dispositivo infectado puede realizar una búsqueda de dispositivos con el bluetooth abierto en su radio de acción y, de forma automática, intentar la transmisión de un fichero infectado. Si se logra la ejecución del fichero infectado en el terminal de la víctima, dicho terminal quedará, a su vez, infectado.

- **NFC**: Las nuevas tecnologías implantadas en los dispositivos también pueden ser susceptibles de sufrir ataques. NFC presenta una debilidad a partir de la cual puede sufrir un ataque man-in-the-middle[1]. A el atacante le basta con disponer de una antena que permita la captura de la transmisión y encontrarse a una distancia lo suficientemente cercana para interceptarla.

1.3.2 Riesgos sobre dispositivos Android

Para llevar a cabo un ataque sobre un dispositivo Android, un atacante intenta explotar toda una serie de riesgos y vulnerabilidades que se dan con mayor o menor frecuencia en los dispositivos móviles. La lista de los 10 riesgos más graves sobre equipos Android es la siguiente [36]:

- **Controles escasos de conexión con el servidor**: Se da cuando una aplicación hace uso de un servicio web o una llamada a una API para establecer una conexión, pero se emplean técnicas de codificación inseguras que dan lugar a toda una serie de posibles vulnerabilidades derivadas de riesgos web. De esta forma, un atacante puede alimentar las entradas de la aplicación con cadenas maliciosas que desencadenen eventos no controlados sobre el servidor.
- **Almacenamiento inseguro**: Ocurre cuando una desarrolladora considera que un usuario o un atacante no va a obtener acceso al sistema de archivos del dispositivo, de manera que se almacena información sensible sin cumplir con los requisitos de seguridad necesarios. De esta forma, un atacante que tenga acceso sobre un dispositivo con permisos de *root* podrá consultar toda la información de las aplicaciones de manera sencilla y sin restricciones.
- **Protección insuficiente en la capa de transporte**: Es frecuente que las aplicaciones utilicen protocolos SSL/TLS para proteger el tráfico que sale del dispositivo durante el proceso de autenticación, pero a menudo dicha protección se limita únicamente a ese momento, permitiendo que todo el resto del tráfico viaje en claro. Esto puede poner en riesgo tanto la información que se está intercambiando como los identificadores de sesión.
- **Fuga de información involuntaria**: Se produce cuando un desarrollador, de manera involuntaria, almacena información sensible en una localización del dispositivo a la que tienen acceso otras aplicaciones del terminal. Generalmente ocurre cuando la aplicación

[1] Ataque en el que un tercero adquiere la capacidad de leer, insertar y modificar los mensajes entre dos partes sin que ninguna de ellas conozca que el enlace entre ellos ha sido violado.

desarrollada está procesando la información introducida por el usuario. Si dicha información es almacenada, aunque sea de manera temporal, haciendo uso de funciones básicas del sistema operativo Android, es probable que ésta pueda ser accesible por otras aplicaciones o servicios ya que se esté almacenando en un lugar compartido por todas las aplicaciones.

- **Autorización y autenticación débil**: A diferencia de los servicios web que hacen uso de *tokens* de sesión para validar al usuario desde que hace login en el sistema, las aplicaciones para dispositivos móviles tienen que estar preparadas para funcionar si éste no dispone de conexión, de manera que deben disponer de toda la funcionalidad para todos los tipos de usuarios integrada en la aplicación, para luego restringir su uso en función de los privilegios de que disponga cada tipo de usuario. Si dicha aplicación no está debidamente protegida, un atacante puede modificar las variables donde se define su rol, o incluso intentar ejecutar la funcionalidad a la que no debe tener acceso saltando el proceso de autenticación.
- **Algoritmos criptográficos vulnerables**: Se da cuando se hace uso de un algoritmo criptográfico cuya baja eficiencia ya ha sido demostrada. Cifrar información sensible de la aplicación haciendo uso de estos algoritmos criptográficos da a un atacante la posibilidad de romper el cifrado y acceder a la información en claro que se quiere proteger.
- **Inyección en el cliente**: Consiste en la introducción de cadenas de texto en los campos y formularios a rellenar por el usuario, que pueden ser interpretadas y ejecutadas por el sistema, dando lugar a la alteración de la ejecución normal del programa. De esta forma, se puede conseguir acceso a privilegios o funcionalidad que el usuario no debería disponer. La ejecución del ataque es similar a la inyección de código sobre aplicaciones web.
- **Decisiones de seguridad a partir de entradas no confiables**: Una aplicación móvil puede recibir datos de diversas fuentes dentro del propio dispositivo. De esta forma, se asegura la interacción entre aplicaciones. Para ello se hace uso de un mecanismo llamado *comunicación entre procesos* o IPC (Inter Process Communication). En el desarrollo de una aplicación, dichas entradas deben ser filtradas, limitando la interacción únicamente con software adicional con el que se quiera intercambiar información. En caso contrario, se puede estar exponiendo información confidencial de que haga uso la aplicación a su captura por otras aplicaciones.

- **Manejo de sesión incorrecto**: Con frecuencia, las aplicaciones para móviles hacen uso de *tokens* de sesión para validar al usuario a lo largo del tiempo una vez iniciada la sesión. Para ello, el servidor sobre el que se ha realizado la autenticación genera una cookie para el usuario, que es guardada por la aplicación y empleada en todas las transacciones que se realicen. El manejo incorrecto de dicho *token* se produce cuando éste es compartido con un atacante durante la comunicación entre cliente y servidor.
- **Ausencia de protecciones binarias**: Todo ejecutable desarrollado por una empresa debe ser protegido haciendo uso de una serie de medidas que eviten la posibilidad de practicar ingeniería inversa sobre dicho software. La ausencia de dichas medidas ofrece a un atacante la posibilidad de decompilar, analizar, modificar y recompilar el programa original. Por un lado, el atacante podrá utilizar partes del programa para desarrollar nuevos programas, incorporando funcionalidad del programa original. Por otro, podrá modificar el programa original, añadir funcionalidad dañina y volver a publicarlo, sin aparente diferencia para los usuarios que hagan uso del mismo.

1.3.3 Protección de la información y seguridad

Para garantizar la seguridad de un dispositivo Android, de manera que se pueda utilizar de forma confiable, se deben conocer, no sólo las posibles amenazas o posibles ataques a realizar sobre dicho terminal, sino las debilidades y factores que pueden dar lugar a nuevas vulnerabilidades o agujeros de seguridad.

La siguiente lista presenta una serie de factores que afectan de manera directa a la seguridad que ofrecen los dispositivos Android [37]:

- **Desconocimiento por parte del usuario final**: El principal factor y, en gran medida, determinante de la aparición del resto de factores, es el desconocimiento que suele tener el usuario común en lo que a temas de seguridad y protección de la información se refiere. La pérdida del dispositivo o el *shoulder surfing*[2] pueden derivar en una serie de riesgos mucho mayores de lo que un usuario normal puede llegar a saber. Suplantación de identidad o substracción de información personal son sólo algunos de los resultados que se pueden dar de no hacer un uso cuidadoso del dispositivo.

  Para evitar estas situaciones, el usuario debe seguir una serie de pautas que evitarán, en gran medida, la filtración de información:

[2] Consiste en observar lo que otro usuario escribe en su dispositivo. Suele darse en lugares públicos como en una cafetería o en el tren. Resulta especialmente grave sobre Smartphones, ya que generalmente resaltan las teclas que presiona el usuario, de manera que se pueden obtener contraseñas, datos de acceso, patrones de desbloqueo o PINS de aplicaciones y dispositivos.

- Cifrado de la información sensible almacenada en el dispositivo o, en caso de ser posible, del propio dispositivo completo.
- Utilización de una contraseña/patrón/PIN para desbloquear el dispositivo.
- Iniciar sesión sobre cada una de las aplicaciones que se vayan a utilizar y cerrar sesión cuando se haya finalizado su uso. No dejar nunca las sesiones abiertas en el dispositivo.
- Proteger el dispositivo de miradas ajenas cuando se introduzcan datos de acceso, o hacer uso de un protector de pantalla.

El seguimiento riguroso de estas medidas evitaría gran parte de los problemas derivados del desconocimiento de los medios de seguridad por parte del usuario.

- **Conexión a redes no seguras**: Los dispositivos actuales cada día permiten la comunicación a través de un mayor número de tecnologías. Desde el bluetooth, implementado en todos los dispositivos desde hace años, hasta el reciente NFC, pasando por tecnologías ampliamente difundidas y utilizadas como la conectividad WiFi. Pero tal como se ha comentado, estas tecnologías son susceptibles de sufrir ataques man-in-the-middle, sniffing[3] o similar, que pueden dar lugar a la visualización, por parte de un tercero, de la información enviada desde el dispositivo.

  Para evitar la monitorización de las conexiones, se debe comprobar que éstas se encuentran autenticadas. De la misma forma, el cifrado del tráfico que se envía a través de redes abiertas permite mitigar, de manera sustancial, la posibilidad de sufrir un ataque de este tipo.

- **Uso de aplicaciones no verificadas**: Tal como se ha comentado, Android permite la instalación de software adicional por parte del usuario. Dicho software puede ser descargado de Google Play, la tienda oficial de aplicaciones de Android, o de otras tiendas como Amazon Appstore. Aunque las aplicaciones ofrecias en dichas tiendas han superado un mínimo control, esto no asegura que se encuentren libres de malware. Por otro lado, los dispositivos permiten la instalación de aplicaciones directamente descargadas de internet a través de paquetes APK. Dichas aplicaciones no han superado ningún examen antes de ser instaladas en el dispositivo, de manera que pueden ser una fuente grave de riesgos.

  Para evitar las amenazas surgidas de la instalación de aplicaciones, se debe hacer uso de la característica *Verify Apps* de Android, y evitar instalar aplicaciones de terceros

[3] Técnica que consiste en escuchar todo lo que circula por una red, generalmente una red interna.

descargadas de internet en la medida de lo posible. Adicionalmente, el uso de un antivirus o programa de monitorización puede ayudar a mantener limpio el dispositivo.

- **Interacción con otros sistemas**: Los dispositivos actuales comparten e intercambian información con toda serie de dispositivos externos. La sincronización de correo, calendarios, agenda, fotos, música y demás. Aunque inicialmente la compartición se solía realizar con un equipo físico, generalmente un ordenador personal, cada día es más común la sincronización de archivos haciendo uso de servicios de la nube.

  Para garantizar la seguridad, se debe considerar la interacción del dispositivo con cualquier agente externo como una fuente de riesgos, de manera que se recomienda la utilización de servicios de sincronización y compartición de archivos únicamente cuando se trata de fuentes confiables y verificadas.
- **Uso de contenido no conocido**: Las tecnologías asociadas a los Smartphones permiten nuevas formas de interacción con elementos externos al móvil, como la lectura de códigos QR. Al igual que en los casos anteriores, dichos elementos pueden suponer una amenaza, con capacidades como la de dirigir al dispositivo de manera automática a una web maliciosa o la descarga de malware en el dispositivo.

  Para evitar estas amenazas, primordialmente el usuario debe estar concienciado y hacer uso del sentido común, evitando así la interacción con elementos no conocidos o de dudosa procedencia. Adicionalmente, el uso de un antivirus puede ayudar a preservar la integridad del dispositivo.
- **Uso de servicios de localización**: Aunque originalmente el GPS del dispositivo estaba más orientado a la funcionalidad de navegación, cada día es más común encontrar aplicaciones que hacen uso de dicha tecnología para ofrecer servicios adicionales a los usuarios. Aunque dichos servicios puedan resultar de utilidad para el usuario, se está atacando la privacidad del mismo, recabando información de los lugares visitados y los movimientos realizados por éste. Adicionalmente, aplicaciones maliciosas que hagan uso de servicios de localización pueden organizar ataques sobre el dispositivo basados en su posición, o recabar información de ámbito personal sobre el usuario con otros objetivos, como su posterior venta.

  Para evitar los ataques basados en el posicionamiento del dispositivo basta con desactivar la localización del mismo en el panel de ajustes. De la misma forma, se puede activar de manera puntual para hacer uso de servicios concretos, como aplicaciones de navegación, aunque se recomienda desactivarlo en cuanto finalice su utilización. A su vez, se recomienda no hacer uso de los servicios de localización de aplicaciones no estrictamente orientadas a ello.

El seguimiento de las recomendaciones anteriormente mencionadas garantiza un nivel de seguridad alto para cualquier usuario no consciente de los riesgos e implicaciones de seguridad de un dispositivo móvil actual. Huelga decir que no existe el mecanismo de protección inexpugnable, de manera que nunca se está absolutamente protegido frente a ataques, pero el conocimiento y la preparación son los primeros pasos para dificultarlos en gran medida.

1.4 Autenticación de dos factores

Dado que es imposible conocer, a ciencia cierta, si un dispositivo con el que se está trabajando se encuentra infectado o no, resulta recomendable tomar medidas adicionales para evitar el acceso ilegítimo a la información con la que se está trabajando.

En este aspecto, se han realizado múltiples estudios, algunos de los cuales quieren proponer sistemas alternativos al empleo de contraseñas para verificar a los usuarios, y otros proponen la utilización de sistemas adicionales a las contraseñas para suplir las deficiencias de éstas. La autenticación de un usuario sobre un sistema se basa en demostrar que éste es quien dice ser. Para ello, se puede hacer uso de tres factores diferenciados, que identifican a la persona de manera única sobre el sistema [38]:

- **Factores de conocimiento**: Algo que el usuario conoce.
- **Factores de propiedad**: Algo que el usuario tiene.
- **Factores inherentes**: Algo que el usuario es.

En el primer caso se encuentran las contraseñas de acceso, conocidas solamente por el usuario que se dispone a hacer login. En el segundo caso se sitúan los *tokens* de acceso o tarjetas magnéticas. En el tercer caso se da el uso de la retina o la huella dactilar para identificar al usuario.

La idea original, y que se implementa hasta la actualidad en la mayor parte de sistemas de autenticación, se basa en la utilización de un único sistema de los anteriormente expuestos para verificar la identidad del usuario, pero dada la enorme cantidad de amenazas que se detectan en la actualidad, se está empezando a hacer uso de la combinación de dos de los sistemas anteriores para garantizar la seguridad del sistema [39]. Este mecanismo es conocido bajo el nombre de *Autenticación de dos factores*, y dado lo tedioso de su implementación, es utilizado en entornos donde prima la seguridad, como la banca online (aún no se han dado casos donde para hacer login en una red social se soliciten dos mecanismos de identificación).

1.4.1 Protocolos de comunicación como soporte al uso de contraseñas

Una de las formas más comunes de implementación de la autenticación de dos factores es la utilización de una pantalla de login estándar sobre un equipo fijo, como puede ser el ordenador personal del usuario, y un mecanismo de doble verificación que haga uso del Smartphone de éste. Dadas las enormes capacidades que tiene un terminal móvil actual, éste mecanismo se puede implementar de múltiples formas.

Algunas de las tecnologías y herramientas con las que cuenta un Smartphone y que pueden ser utilizadas a tal fin, son las siguientes:

- **Bluetooth**: Se trata de un protocolo de comunicaciones basado en radiofrecuencia que permite establecer una comunicación entre dos dispositivos a corto alcance. Tiene un índice nominal de 10 m, y las tasas de transferencia de datos llegan hasta 1 Mbps [40]. Dado el bajo consumo de esta tecnología y la ineroperabilidad entre dispositivos al margen de tratarse de marcas distintas, se emplea comúnmente para establecer una conexión o la transmisión de información entre pequeños dispositivos.
- **NFC**: Consiste en un sistema de comunicación inalámbrico de corto alcance y frecuencias de transmisión altas. Su funcionamiento se basa en la generación de un campo electromagnético que utiliza para comunicarse con otros elementos, y su objetivo principal es la identificación y validación de dispositivos de manera rápida y eficaz. Posee una tasa de transferencia de 424 kbps y su rango de funcionamiento máximo es de 20 cm, pero a diferencia del Bluetooth, permite obviar todo el procedimiento de activación, escaneo de dispositivos y confirmación de la conexión [41] [42].
- **Cámara**: Determina un canal de comunicación unidireccional del exterior con el dispositivo. Permite realizar capturas de la información observada, de manera que puedan ser procesadas de manera digital, incluyendo el reconocimiento de textos, patrones, matrices o códigos (como los códigos de barras o los códigos QR):
    - **QR**: Es una matriz de puntos que permite almacenar información. Su objetivo inicial era facilitar la administración de inventarios, aunque al introducir en los Smartphones la capacidad de leer el contenido de estos códigos, su popularidad ha aumentado, creciendo de la misma manera el número de usos que se les puede dar. Presenta tres elementos, que deben estar presentes en todos los códigos QR en la misma colocación, y son los encargados de posicionar, alinear y sincronizar el código. El resto de elementos es variable, y es donde se concentra el grueso de información codificada [43].

Aunque un Smarphone actual dispone de más tecnologías que podrían llegar a adaptarse para su utilización con el objetivo de verificar al usuario, las anteriores han demostrado su correcto funcionamiento y adaptabilidad a la hora de cumplir dicho objetivo.

1.4.2 Escalado de privilegios

Una de las principales brechas de seguridad que presenta Android es la posibilidad de lograr un escalado de privilegios haciendo uso de vulnerabilidades del sistema. Aunque dichas vulnerabilidades van siendo arregladas, dispositivos antiguos o desactualizados pueden encontrarse en riesgo.

Esta vulnerabilidad proporciona a un atacante la posibilidad de actuar sobre el sistema como superusuario, obteniendo acceso a información confidencial o funcionalidad restringida.

La combinación de un escalado de privilegios con un ataque de plataformas cruzadas, donde el atacante controla una red y puede suplantar al terminal móvil o al servidor de acceso, puede suponer un grave riesgo para la información transmitida, dado que por defecto se confía en el destinatario de la información, y la vulnerabilidad viene dada a nivel de sistema operativo, y no de la propia aplicación [44].

El efecto de este tipo de ataque puede verse mitigado al cifrarse la información que se envía a través de la red y ser procesada directamente por el destinatario, aunque no hay garantías absolutas para este tipo de ataque.

Estudio inicial del sistema

El estudio inicial del sistema se emplea para obtener una idea general de la amplitud y las necesidades del sistema antes de proceder con su desarrollo. Se realiza una valoración del alcance del proyecto de manera rápida y superficial, permitiendo una valoración del equipo necesario para llevar a cabo la implementación del sistema. A su vez, se proponen distintas alternativas de desarrollo para cada uno de los módulos que componen el sistema, seleccionando las que resulten más óptimas y proporcionando una explicación detallada de las razones que han llevado a decantarse por una alternativa concreta.

2.1 Introducción al sistema

La evolución de las tecnologías en la actualidad está dando lugar a un crecimiento tanto de la capacidad de los equipos como de los ataques que se pueden llevar a cabo. Estrategias de ataque que hace años solamente podían concebirse en la teoría, pueden ponerse ahora en práctica de manera sencilla y asequible para toda clase de usuarios, además de la enorme cantidad de vulnerabilidades y nuevas formas de infección o ataque que se descubren día a día dada la enorme extensión que está viviendo la informática.

Por otro lado, los esfuerzos no se centran únicamente en el diseño de nuevas formas de ataque más complejas o más destructivas. Una parte del sector de desarrollo se encarga de la creación de nuevos kits de implantación[4], que cada día hacen más fácil y más asequible la realización de un ataque. De esta forma, anteriormente un atacante debía disponer de amplios conocimientos de informática para la realización de un ataque de forma satisfactoria, generalmente compilando sus propios programas para la ejecución del mismo. Pero hoy en día todo el proceso se ha simplificado hasta el punto de que cualquier usuario, con unos conocimientos de

[4] Software como Metasploit está orientado al desarrollo y ejecución de exploits.

informática mínimos, puede descargar las herramientas necesarias para intentar atacar un servidor público, romper la contraseña de un sistema seguro o generar un archivo infectado y publicarlo online para su descarga, infectando a cientos de equipos.

2.1.1 La necesidad de un sistema seguro

Toda esta situación está dando lugar a que cada día resulte más difícil conocer si nuestro equipo personal se encuentra infectado o no. Y un equipo infectado con un troyano o un keylogger puede poner en bandeja a un atacante toda la información manejada por el usuario. Por un lado, las capturas de pantalla o la cámara pueden violar la intimidad del usuario, ofreciendo al atacante toda la información visualizada por el usuario o, incluso, fotos de éste. Por otro lado, el seguimiento de teclas pulsadas ofrece acceso a la poca información que puede aparecer codificada en la pantalla, como contraseñas, PINs, claves de acceso a cuentas bancarias y demás información sensible.

La utilización de software antivirus, antispyware, firewall y demás puede proteger el equipo de manera efectiva, pero nunca de manera infalible. La investigación y la aparición de nuevos ataques día a día que aún no han sido registrados en las bases de datos de dicho software los vuelve indetectables.

Por ello, se hace necesario emplear mecanismos de seguridad y verificación alternativos o adicionales para garantizar la seguridad de un sistema.

2.1.2 La utilización de sistemas adicionales a las contraseñas

Tal como se ha explicado hasta ahora, la contraseña es un sistema de seguridad obsoleto, fácilmente atacable y con una serie de requisitos para el usuario que resultan sumamente difíciles de cumplir. Generalmente, la razón por la que se sigue utilizando este sistema para proteger las cuentas de usuario es su ínfimo precio de implementación y su extensión. Hoy en día, todo usuario conoce el funcionamiento de una contraseña, mientras que sistemas alternativos deberían ser implantados y explicados, con la consiguiente reticencia por parte del usuario final, reacio a cambiar el sistema que conoce.

Pero dada la facilidad actual para detectar la contraseña de un usuario, hay sistemas donde se debe aplicar un nivel de seguridad mayor para garantizar la impenetrabilidad del sistema. Sistemas como una red social contienen información personal referente al usuario, pero dicha información no resulta tan sensible como, por ejemplo, un sistema bancario. Además, la utilización de mecanismos alternativos a la contraseña supondría una implicación mayor por parte del usuario y un consiguiente aumento del tiempo para hacer login, lo cual en sistemas,

como una red social, resulta inconcebible, dado el alto número de conexiones que puede llegar a realizar un usuario normal por día. Por el contrario, se trata de un sistema fácilmente implantable, por ejemplo, en entornos bancarios, dado que el usuario agradece los sistemas de seguridad adicionales en lo que se refiere a la protección de sus ingresos o ahorros.

Tal como se ha comentado anteriormente, la utilización del Smartphone permite la combinación de la contraseña con sistemas adicionales de verificación de la identidad. Dado que se trata de un dispositivo cada día más extendido, se puede concluir como una de las mejores opciones para verificar al usuario. De esta forma, en el momento en que el usuario acceda al sistema o realice alguna acción que requiera de su confirmación, se puede solicitar que, además de introducir la contraseña en el sistema, realice alguna acción con su terminal móvil para demostrar que él es quien ha solicitado la realización del comando seleccionado. En caso de que se trate de una acción fraudulenta llevada a cabo de forma remota por un atacante que ha conseguido acceso al equipo de la víctima o a sus datos de acceso, al no disponer del dispositivo móvil de ésta, no podrá confirmar la acción, de manera que esta no será validada por el sistema y no se llevará a cabo.

2.2 Alcance del sistema

El sistema a desarrollar pretende simular un entorno bancario que permite la realización de transferencias de una cuenta a otra, empleando los sistemas de verificación de la identidad del usuario comentados anteriormente. Para ello, previamente se llevará a cabo un estudio donde se analice cada una de las opciones a implementar de manera detallada, seleccionando la opción a llevar a cabo en la versión final del sistema.

El proyecto se compone de dos aplicaciones. La primera consiste en una aplicación web con acceso controlado por usuario y contraseña (DNI y PIN), similar a la empleada por todos los bancos que ofrecen funcionalidad online. Sobre dicha aplicación, el usuario puede consultar su saldo y sus movimientos, y tiene la opción de realizar una transferencia a otra cuenta. En el momento en que se solicita esta última opción, se solicita su confirmación a través de la aplicación móvil.

Esta segunda aplicación, desarrollada para dispositivos Android, permite al usuario confirmar la transferencia que acaba de solicitar. El mecanismo empleado para confirmar la transferencia y la tecnología a utilizar serán detallados posteriormente, cuando se haya realizado la valoración de las distintas alternativas y se haya seleccionado la opción más conveniente dadas las características del proyecto.

2.3 Alternativas de solución

En el momento de estructurar el sistema se plantean varias alternativas que permiten alcanzar la misma funcionalidad a través de medios distintos, de manera que se realiza un estudio detallado de cada una de las opciones planteadas y se procede a seleccionar aquellas que resulten más eficientes para la implementación y posterior funcionamiento de la aplicación.

Se realiza un estudio de cada uno de los subsistemas para los cuales se plantean alternativas, de manera que se puede realizar una valoración crítica de manera independiente para cada una de las propuestas. A continuación se ofrece un resumen del estudio realizado, acompañado de las conclusiones extraídas y la alternativa seleccionada.

2.3.1 Tecnología a emplear

El sistema que se va a desarrollar se compone, por un lado, de una aplicación web que permite al usuario autenticarse y realizar una transferencia, y por otro, de una aplicación para dispositivos móviles que verifica la identidad del usuario. Para llevar a cabo dicha autenticación, se plantean diversos mecanismos que pueden validar la identidad de éste, pero su funcionamiento e implementación varía en gran medida, de manera que se realiza un breve estudio del proceso de implementación que supone cada uno de ellos.

2.3.1.1 Bluetooth

Se trata de un protocolo que permite la comunicación, de manera eficaz, entre dispositivos que se encuentren cerca unos de otros. Sus tasas de transmisión son altas y su consumo de recursos es muy bajo. Su implantación en el sistema permitiría la comunicación entre el equipo fijo del usuario (ordenador personal) y su Smartphone, permitiendo validar que éste es quien ha realizado una transferencia en el momento en que el comando sea ejecutado. Dado que el dispositivo de comunicación Bluetooth del equipo consiste en un recurso hardware, y no se puede acceder a dichos recursos desde un explorador web, la implementación del sistema pasaría por el desarrollo de una pequeña aplicación de escritorio que permite la comunicación con el dispositivo Bluetooth. Dicha aplicación debería ser instalada por el usuario la primera vez que se pretende acceder a la web desarrollada. De la misma forma, se deberá tener instalada la aplicación para Android que permite la confirmación de las operaciones. Para verificar la identidad del usuario, tanto el equipo como el dispositivo móvil deben tener el Bluetooth encendido y se debe haber establecido una comunicación entre uno y otro. Una vez hecho esto, el usuario hace login en la web con su nombre de usuario y contraseña, accede al subsistema de transferencias y solicita una transferencia de una cantidad de dinero concreta a otra cuenta. En el momento en que se recibe el comando que solicita la transferencia, el subsistema web,

haciendo uso del módulo que permite la comunicación con el dispositivo Bluetooth, establece la comunicación con el dispositivo móvil. Éste, que posee el identificador del usuario, haciendo uso de la aplicación móvil devuelve dicho identificador al equipo, que confirma que el usuario es quién dice ser y confirma la transferencia solicitada.

2.3.1.2 NFC

Es un sistema de comunicación entre dispositivos que se basa en la proximidad entre estos. El proceso de implantación en el sistema, al igual que en el caso del bluetooth, requiere de la instalación de un pequeño software que permita el acceso al recurso desde un explorador web. Adicionalmente, dado que los equipos actuales no incluyen transmisores NFC, se hace necesaria la instalación de un dispositivo externo para establecer la comunicación.

El procedimiento de autenticación para el usuario final es similar al caso anterior. Éste debe haber conectado el dispositivo externo NFC al equipo y haber instalado el pequeño software necesario para permitir la comunicación entre la aplicación web y el dispositivo hardware. Una vez hecho esto, se hace login en la página web y se solicita ejecución de una transferencia. La aplicación web, haciendo uso del software instalado por el usuario en el equipo, se comunica con el dispositivo móvil a través de NFC, siempre que éste se encuentre a 20cm o menos del equipo. El Smartphone, a través de la aplicación instalada, devuelve al ordenador el identificador del usuario, demostrando que, efectivamente, éste es quien ha solicitado la ejecución del comando, confirmando la transferencia.

Tal como se puede observar, se obvia el paso en el que se requiere la activación del puerto y la identificación manual de los dispositivos, lo cual supone una mejora considerable respecto al Bluetooth.

2.3.1.3 QR

A diferencia de los dos anteriores, QR no es un protocolo de comunicación, sino que consiste en un mecanismo de codificación de la información. Por ello, el procedimiento de autenticación del usuario varía bastante.

Para hacer uso del sistema, el único prerrequisito que se aplica es la necesidad de haber instalado la aplicación para Android. Una vez hecho esto, el usuario accede a la web y hace login igual que en los casos anteriores. Cuando se solicita la realización de una transferencia, el usuario recibe un código QR en la pantalla que deberá escanear con su aplicación móvil. El conjunto de la cadena contenida en el QR y los datos del usuario contenidos en el dispositivo generan un nuevo código. Dicho resultado será enviado al servidor de la aplicación, donde en caso de haberse recibido un código válido, se confirmará la transferencia.

Se puede apreciar que este procedimiento no requiere de la instalación de software adicional en el equipo ni la adquisición de ningún hardware externo, aunque la interacción necesaria por parte del usuario es algo mayor.

2.3.1.4 Comparativa

Dado que las tres opciones resultan ser alternativas factibles de ser aplicadas en el sistema final, se procede a realizar una comparativa donde se exponga un listado de las ventajas y desventajas que presentan cada una de ellas:

Bluetooth	
Ventajas	Desventajas
• **Rapidez de transmisión.**	• Requiere la habilitación manual del Bluetooth. • Requiere la conexión manual a los dispositivos. • Necesaria la instalación de un pequeño software en el equipo. • En caso de equipos sin Bluetooth, requiere de hardware adicional.

Tabla 2-1: Ventajas y desventajas del uso de Bluetooth

NFC	
Ventajas	Desventajas
• **Procedimiento automático.** • **Baja participación del usuario.**	• Muy corto alcance. • Limitado a Smartphones con NFC. • Necesario hardware adicional para el equipo. • Necesaria la instalación de un pequeño software en el equipo.

Tabla 2-2: Ventajas y desventajas del uso de NFC

QR	
Ventajas	Desventajas
• **No requiere de la instalación de software adicional.** • **No requiere de la utilización de dispositivos externos.**	• Mayor implicación por parte del usuario. • Necesaria conexión del dispositivo móvil con internet.

Tabla 2-3: Ventajas y desventajas del uso de QR

Tal como puede observarse, cada una de las opciones presenta una serie de ventajas y desventajas que claramente las identifica y diferencia de las demás alternativas. En un primer

lugar, el uso de Bluetooth, dada la antigüedad de este sistema, implica una serie de configuraciones y puesta en funcionamiento que puede suponer una molestia para el usuario. Por el contrario, el uso de NFC, al ser un sistema muy nuevo y poco extendido, puede suponer el alejamiento de gran parte de los usuarios, o la obligación de adquirir nuevos dispositivos para permitir su funcionamiento. Adicionalmente, la obligación de instalar un software adicional al uso de la web puede no ser del agrado de todos los usuarios. Por otro lado, la tecnología QR, al haber vivido un enorme crecimiento con la extensión de los Smartphones, es utilizada día a día por una enorme comunidad de usuarios. Aunque su utilización en el sistema suponga una mayor implicación del usuario en el proceso de autenticación, también puede ser visto como un mecanismo más seguro al requerir una participación activa de éste.

2.3.1.5 Conclusión

Dado que se considera que requerir la instalación de software adicional puede no ser del gusto de todos los usuarios, y que las tecnologías Bluetooth y NFC se encuentran o desfasada o poco extendida, se decide validar al usuario en la versión final de la aplicación haciendo uso de la tecnología QR.

2.3.2 Claves de Seguridad

Para verificar la identidad del usuario, se debe hacer uso de un sistema de claves compartido por la aplicación web y la aplicación móvil, que permita identificarlo de manera única.

Tal como se ha comentado anteriormente, la aplicación web se comunica con la aplicación móvil para comprobar si el usuario que está haciendo uso de la misma es quien dice ser. De esta forma, cuando se solicita una transferencia desde el portal, el usuario debe confirmar dicha transferencia haciendo uso de su Smartphone. El funcionamiento de este sistema se basa en un sistema de claves que se emplea para intercambiar un mensaje entre ambos dispositivos. En primer lugar, la aplicación web cifra un mensaje de confirmación con una clave conocida por la aplicación móvil, que será única para cada usuario y estará almacenada en el Smartphone. Dicho mensaje deberá ser descifrado por la aplicación móvil para confirmar su contenido, tras lo cual se volverá a cifrar y será enviado de nuevo a la aplicación web. Por último, ésta descifra el contenido del mensaje y comprueba si la respuesta de la aplicación móvil es correcta. En caso de que así sea, la identidad del usuario queda confirmada, de manera que se confirma la transferencia.

El procedimiento para llevar a cabo el intercambio de mensajes se puede llevar a cabo de varias maneras. A continuación se plantean las distintas alternativas:

2.3.2.1 Cifrado simétrico

La clave empleada para cifrar y descifrar los mensajes es la misma, de manera que ésta es compartida por ambos dispositivos. El funcionamiento es igual para el cifrado que para el descifrado. A la cadena de texto en claro se le aplica el algoritmo de cifrado, lo cual da lugar a la cadena de texto cifrado. Para descifrar el mensaje se aplica el algoritmo de cifrado a la cadena de texto cifrada, dando lugar al texto original o la cadena de texto descifrado.

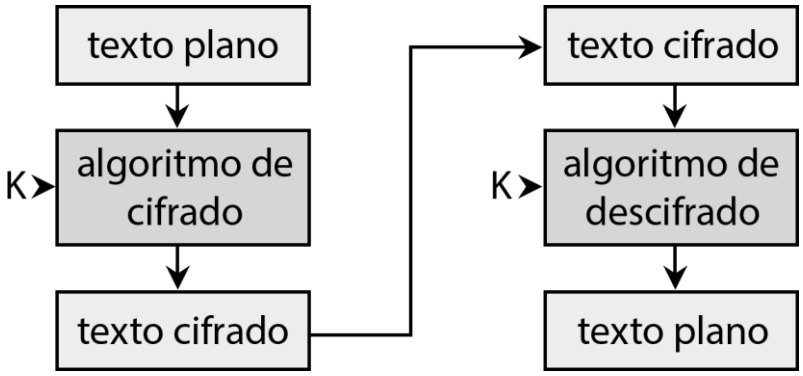


Figura 2-1: Proceso de cifrado simétrico

En caso de hacer uso de cifrado simétrico, se empleará AES, utilizado desde 2001 como estándar de cifrado simétrico desde que quedaran probadas las vulnerabilidades del DES. Su funcionamiento se basa en el uso de redes de sustitución y permutación. En función del tamaño de la clave a emplear, se aplican una serie de rondas, en cada una de las cuales la información sufre una serie de transformaciones:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

2.3.2.2 Cifrado asimétrico

Se trata de un procedimiento de cifrado en el que no se comparte la clave, de manera que las claves de cifrado y descifrado son distintas. De esta forma, cada una de las entidades que intercambian los mensajes dispone de un juego de claves propio:

- **Clave pública**: Es la clave que se da a conocer a las entidades con las que se quiere entablar un intercambio de mensajes. Su conocimiento les permite el cifrado de mensajes, y resulta imposible deducir la clave privada de un usuario a partir de la clave pública.

- **Clave privada**: Es la clave secreta conocida únicamente por la entidad que recibe el mensaje. Su conocimiento permite descifrar los mensajes cifrados con la clave pública, de manera que no debe ser revelada.

Dichas claves se emplean para realizar operaciones complementarias, y permiten tanto el cifrado de información como la generación y verificación de firma. La principal ventaja que presenta este sistema respecto al cifrado de clave simétrica es que se evita el paso del intercambio de claves.

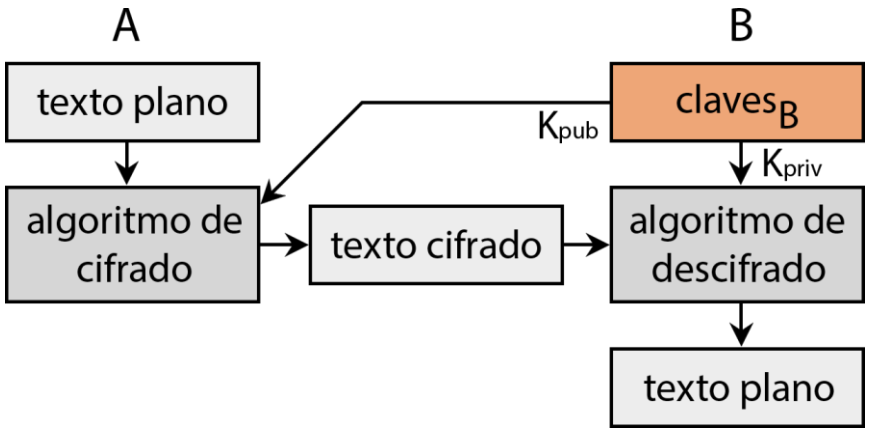


Figura 2-2: Proceso de cifrado asimétrico

En el caso del cifrado asimétrico, se haría uso de RSA, diseñado en 1977. Se trata de una función de cifrado basada en descomposición por bloques, y su resistencia viene determinada por la alta complejidad del uso de la potenciación modular.

2.3.2.3 Comparativa

Dado que ambas opciones resultan factibles de aplicarse satisfactoriamente en el sistema, se realiza un estudio de ventajas y desventajas para cada una:

Cifrado simétrico	
Ventajas	Desventajas
• **Cifrado de mensajes muy rápido.**	• Dificultad para realizar el intercambio de claves. • El servidor debe disponer de una clave distinta para cada usuario.

Tabla 2-4: Ventajas y desventajas del uso de cifrado simétrico

Cifrado asimétrico	
Ventajas	Desventajas
• **Más seguro desde el punto de vista del criptoanálisis.** • **Posibilidad de cifrar y firmar mensajes.**	• Mayor carga computacional.

Tabla 2-5: Ventajas y desventajas del uso de cifrado asimétrico

Se aprecia una alta igualdad entre ambos sistemas. Por un lado, el cifrado simétrico ofrece un tiempo de procesamiento mucho más rápido que el cifrado asimétrico, aunque en el proyecto actual el tiempo de cifrado no supone un factor clave, ya que se realiza un cifrado de mensaje que se envía del subsistema web a la aplicación móvil, y un segundo cifrado que se envía en sentido contrario como respuesta, de manera que un mayor tiempo de procesamiento no supone un problema. Por otro lado, el intercambio de claves sí que puede suponer un problema, aunque se podría realizar en el momento de activación de la aplicación móvil en la propia entidad bancaria.

Por otro lado, el cifrado asimétrico ofrece la posibilidad de firmar los mensajes, aunque dicha capacidad no había sido considerada en un principio, se le puede sacar provecho más adelante. Adicionalmente, el cifrado asimétrico es considerado más seguro desde el punto de vista del criptoanálisis dada su alta complejidad, que también es la razón por la cual su carga computacional es mayor. Por último, se evita el factor del intercambio de clave al disponer de claves pública y privada para cada entidad, de manera que para cifrar un mensaje y que solamente lo pueda descifrar el receptor basta con cifrarlo con la clave pública de éste.

2.3.2.4 Conclusión

Dados los factores expuestos anteriormente, se decide hacer uso de cifrado asimétrico para cifrar los mensajes que se intercambian el servicio web y la aplicación móvil, ya que su utilización dota de mayor seguridad al sistema y no se hace necesario el intercambio de claves, como ocurre en el cifrado simétrico.

Análisis

El análisis del sistema se emplea para detallar las especificaciones formales que describen el sistema propuesto. Con el alcance del sistema ya definido, se realiza una descripción detallada del sistema, a partir de la cual se determinan los requisitos que darán lugar, tanto a las acciones que se pueden desarrollar sobre el sistema, como a los objetivos que se deben cumplir. De la misma forma, se definen los diagramas de actividades y se determina el catálogo de casos de uso. El objetivo es dejar el sistema definido antes de proceder a la fase de Diseño del mismo.

3.1 Descripción general

El sistema se compone de dos aplicaciones diferenciadas que interactúan la una con la otra y cuya utilización conjunta da sentido al proyecto. El objetivo es el de dotar de la seguridad necesaria a un sistema crítico como es un modelo bancario de transferencia de fondos entre cuentas, donde se considera que una configuración de usuario y contraseña no proporciona el nivel de seguridad mínimo dada la facilidad para perpetrar un ataque sobre dicho sistema.

3.1.1 Aplicación web

La aplicación web ofrece a un usuario la posibilidad de hacer login en el sistema con sus datos de acceso. En caso de que dicho usuario no se encuentre registrado en el sistema, éste deberá darse de alta. La pantalla de registro solicita un nombre de usuario y una contraseña para el posterior acceso al sistema, nombre y apellidos del usuario, DNI, fecha de nacimiento, dirección, teléfono y email. Adicionalmente, la base de datos donde se registra la información del usuario dispone de un campo *identificador del teléfono*, donde se almacena el identificador del mismo al acceder a la aplicación móvil por primera vez, así como un número de cuenta único. Hasta que no se haya formalizado el registro de manera completa, incluyendo la instalación de la aplicación móvil, el usuario no podrá acceder al sistema.

Una vez formalizado el registro, el usuario puede ingresar en el sistema haciendo uso de la pantalla de login. Para ello debe introducir sus datos de acceso, tras lo cual se le solicitará el DNI como mecanismo de seguridad adicional. Una vez completados ambos pasos, el usuario se encuentra en su página principal personal, desde donde tiene acceso a la siguiente funcionalidad:

- **Movimientos**: Ofrece un listado por orden cronológico de los pagos y cobros de dinero realizados sobre la cuenta del usuario.
- **Transferencia**: Permite al usuario realizar un movimiento de dinero de su cuenta personal a la de otro usuario. En una primera pantalla se solicitan los datos de la cuenta sobre la que se quiere realizar el ingreso y la cantidad de dinero a ingresar. Para confirmar la realización de la transferencia, se ofrece una segunda pantalla en la que se muestra un código QR que el usuario debe escanear haciendo uso de su Smartphone.
- **Datos personales**: Muestra al usuario los datos con los que se registró en el sistema. En caso de ser necesario, se le ofrece la opción de editarlos.

3.1.2 Aplicación móvil

Para hacer uso del sistema completo, el usuario debe completar el registro de sus datos mediante la aplicación para Android. Para ello, en el momento en que ha completado el registro en el portal web, se le solicita la instalación de la aplicación para Smartphone, donde al ingresar con su nombre de usuario y su contraseña, se almacena el identificador del teléfono en la base de datos de la aplicación. Dicho identificador se utilizará para generar los códigos QR de confirmación, de manera que únicamente puedan ser descifrados por dicho terminal. Una vez finalizado el registro, la aplicación accede directamente a la pantalla principal, donde se ofrece al usuario la posibilidad de realizar la captura de un código QR. En el momento en que el portal web solicite al usuario la confirmación de una transferencia, éste hará uso de dicha funcionalidad, lo cual ofrece una cámara con la que se debe enfocar el código mostrado por pantalla. Cuando se realice la captura del código, se procesa el contenido del mismo y se envía la respuesta al servidor. En caso de confirmarse el envío realizado por la aplicación, se mostrará un mensaje de éxito. En caso contrario, se informará al usuario que la transferencia no ha podido confirmarse.

3.1.3 Panel del Administrador

Adicionalmente a las aplicaciones disponibles para los usuarios finales, se ofrece funcionalidad de administrador para monitorización de actividades, orientada a su uso por parte de la entidad bancaria que ofrece la aplicación.

El administrador puede consultar los datos personales de cualquier usuario, además de tener la capacidad de modificar dichos datos y dar de alta a nuevos usuarios. De la misma forma, puede acceder al listado completo de transferencias realizadas.

Para acceder a la funcionalidad de administrador, éste hace login en la página normal con sus credenciales de acceso. Para limitar el acceso y garantizar la seguridad de dicha cuenta, solamente se permite el acceso a la misma desde la red interna de la empresa, y haciendo uso de la dirección MAC del equipo designado a tal fin.

3.2 Diagrama de interacción entre elementos

El sistema se compone de una aplicación web alojada en un servidor la cual, a su vez, es accedida desde terminales fijos (equipos domésticos) y terminales móviles (Smartphones). La interacción que desarrolla el servidor en cada uno de los casos es diferente, tal como se detalla en la Figura 3-1.

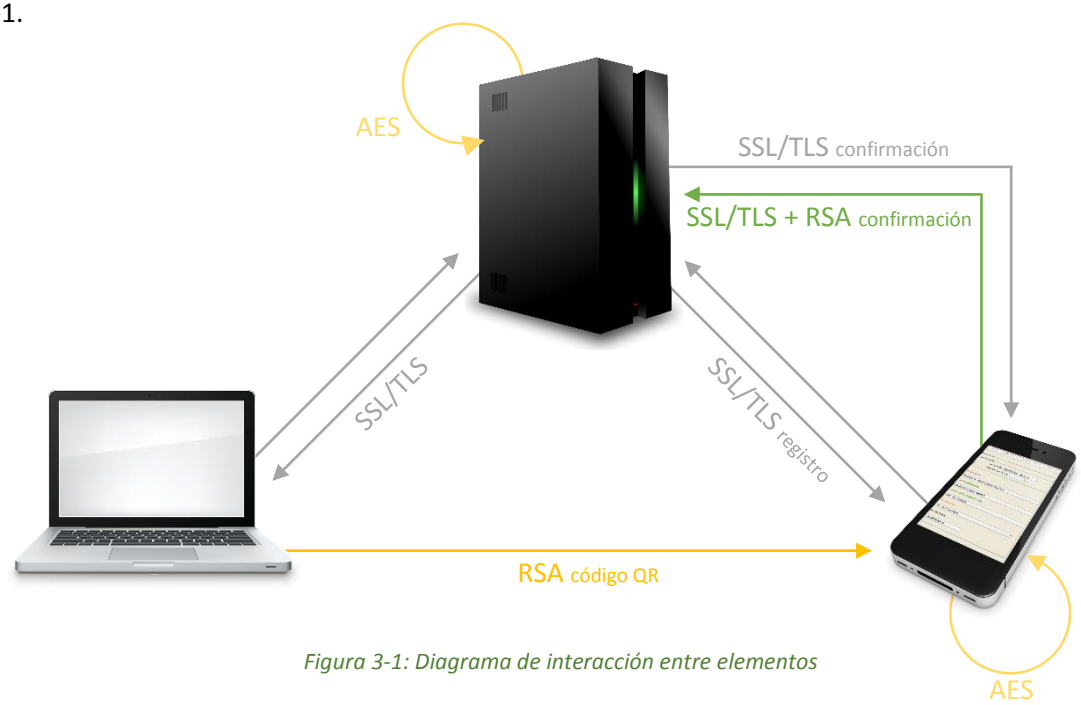


Figura 3-1: Diagrama de interacción entre elementos

3.3 Usuarios

La aplicación será utilizada por dos tipos de usuarios diferenciados:

- **Administrador**: Dispone de control total sobre el sistema. Tiene acceso a la información personal e historial de transferencias de todos los usuarios. Además tiene la capacidad de crear nuevos usuarios y modificar la información de los existentes. Únicamente hace uso de la aplicación web, y su acceso está controlado por contraseña y localización.

- **Usuario final**: Se trata de todo usuario que haga uso de los servicios que ofrece el sistema. Dependiendo del momento en que se encuentre de uso de la aplicación, puede interpretar dos roles diferenciados:
  - **Usuario no registrado**: Es la primera vez que el usuario hace uso del sistema. Sus datos todavía no han sido almacenados en la base de datos de la aplicación, de manera que se le ofrece un formulario de registro.
  - **Usuario registrado**: El usuario ya se encuentra dado de alta, de manera que puede acceder al sistema y hacer uso de todas las funcionalidades ofrecidas, exceptuando las reservadas al administrador.

3.4 Requisitos de usuario

La lista de requisitos de usuario permite conocer los requisitos que, en primera instancia, se pretende que cumpla la aplicación. Cada uno de los requisitos se define de la siguiente manera:

- **Nombre**: Comienza por las iniciales "RU" para indicar que se trata de un requisito de usuario, acompañadas de "RC" para requisitos de capacidad o "RR" para requisitos de restricción. Es seguido por un número que se utiliza para identificar al requisito, y un nombre, que describe de manera breve y general el contenido del requisito.
- **Prioridad**: Nivel de importancia que presenta el requisito. Puede tomar los valores "Alta", "Media" y "Baja".
- **Necesidad**: Permite conocer el nivel de exigencia de implementación. Puede tomar los valores "Esencial" y "Deseable".
- **Claridad**: Determina el nivel de precisión con que está redactado el requisito. Puede tomar los valores "Alta", "Media" y "Baja".
- **Verificabilidad**: Precisa la capacidad para comprobar el cumplimiento de un requisito concreto. Puede tomar los valores "Alta", "Media" y "Baja".
- **Descripción**: Explicación del requisito.

3.4.1 Requisitos de Capacidad

Determinan las acciones que el usuario puede realizar sobre el sistema.

RURC01 – Registro de usuarios			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un usuario puede darse de alta en el sistema.			

Tabla 3-1: Requisito de Capacidad - Registro de usuarios

RURC02 - Login			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un usuario puede hacer login en el sistema con sus datos de acceso.			

Tabla 3-2: Requisito de Capacidad - Login

RURC03 – Consulta movimientos			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un usuario puede consultar los movimientos de su cuenta.			

Tabla 3-3: Requisito de Capacidad - Consulta movimientos

RURC04 – Consulta/modificación datos			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un usuario puede consultar los datos personales con los que se ha registrado en el sistema y modificarlos en caso de que sea necesario.			

Tabla 3-4: Requisito de Capacidad – Consulta/modificación datos

RURC05 – Realiza transferencia			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un usuario puede realizar una transferencia de dinero a otra cuenta.			

Tabla 3-5: Requisito de Capacidad - Realiza transferencia

RUR06 – Confirma transferencia			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un usuario puede confirmar una transferencia haciendo uso de la aplicación móvil.			

Tabla 3-6: Requisito de Capacidad - Confirma transferencia

RURC07 – Volver al menú			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un usuario puede volver al menú principal en cualquier momento.			

Tabla 3-7: Requisito de Capacidad - Volver al menú

RURC08 - Logout			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un usuario puede desconectarse del sistema.			

Tabla 3-8: Requisito de Capacidad - Logout

RURC09 – Consulta información			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un administrador puede consultar los datos personales de cualquier usuario.			

Tabla 3-9: Requisito de Capacidad - Consulta información

RURC10 – Modifica información			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
Un administrador puede modificar los datos personales de los usuarios.			

Tabla 3-10: Requisito de Capacidad - Modifica información

RURC11 – Consulta movimientos			
Prioridad	☐ Alta ☐ Media ☒ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☐ Alta ☒ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un administrador puede consultar el listado completo de transferencias realizadas.			

Tabla 3-11: Requisito de Capacidad - Consulta movimientos

RURC12 – Nuevo usuario			
Prioridad	☐ Alta ☐ Media ☒ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Un administrador puede crear nuevos usuarios.			

Tabla 3-12: Requisito de Capacidad - Nuevo usuario

3.4.2 Requisitos de Restricción

Determinan las condiciones que debe cumplir el sistema.

RURR01 – Identificador en QR			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☐ Alta ☒ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
Se debe incluir información única identificativa del dispositivo en el QR generado para asegurar que solamente se pueda validar desde dicho terminal.			

Tabla 3-13: Requisito de Restricción - Identificador en QR

RURR02 – Hash de contraseña			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
Se debe almacenar únicamente el Hash de la contraseña del usuario.			

Tabla 3-14: Requisito de Restricción - Hash en contraseña

RURR03 – Cifrar BBDD			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
Se debe cifrar el contenido de la base de datos para evitar posibles filtraciones de información.			

Tabla 3-15: Requisito de Restricción - Cifrar BBDD

RURR04 – Evitar ataques web			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☐ Media ☒ Baja
Descripción			
Se debe proteger la página web contra ataques basados en SQL Injection, XSS y demás.			

Tabla 3-16: Requisito de Restricción - Evitar ataques web

RURR05 – Coherencia de diseño			
Prioridad	☐ Alta ☐ Media ☒ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☐ Alta ☒ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
La interfaz de ambas aplicaciones debe guardar coherencia y mantener un patrón de color y diseño uniforme y agradable.			

Tabla 3-17: Requisito de Restricción - Coherencia de diseño

3.5 Modelo de Casos de Uso

El modelo de casos de uso representa, a partir de los requisitos de usuario, cada una de las acciones que el usuario final podrá realizar sobre la aplicación.

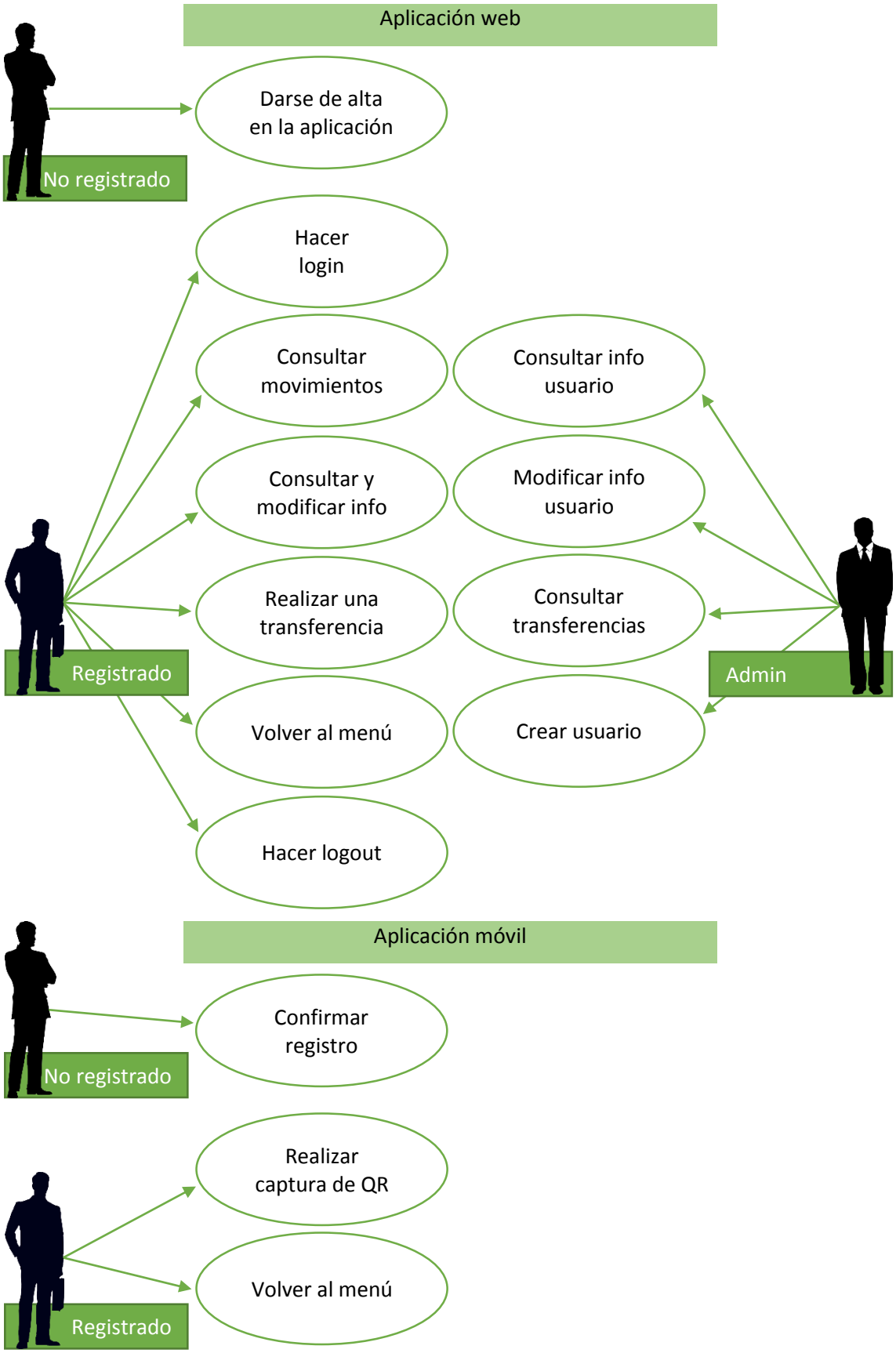


Figura 3-2: Modelo de casos de uso

3.5.1 Aplicación web

Acciones a realizar por el usuario sobre la web del sistema.

Darse de alta en la aplicación	
Actores	Usuario no registrado
Descripción	
El usuario introduce un nombre de usuario, una contraseña, y su nombre, apellidos, DNI, fecha de nacimiento, dirección, teléfono y email para darse de alta en el sistema.	
Precondiciones	El usuario accede por primera vez a la web.
Postcondiciones	La información del usuario es almacenada en la base de datos.

Tabla 3-18: Caso de uso - Darse de alta en la aplicación

Hacer login	
Actores	Usuario registrado
Descripción	
El usuario introduce su nombre de usuario y contraseña en la pantalla de login para acceder al sistema.	
Precondiciones	El usuario se ha dado de alta en el sistema.
Postcondiciones	El usuario se encuentra en el menú principal de la aplicación.

Tabla 3-19: Caso de uso - Hacer login

Consultar movimientos	
Actores	Usuario registrado
Descripción	
El usuario accede a la página donde se muestran los movimientos de fondos realizados desde su cuenta.	
Precondiciones	El usuario se encuentra conectado al sistema.
Postcondiciones	Se muestra la lista detallada de movimientos.

Tabla 3-20: Caso de uso - Consultar movimientos

Consultar y modificar info	
Actores	Usuario registrado
Descripción	
El usuario accede a la página donde se muestran sus datos personales, pudiendo cambiar estos en caso de ser necesario.	
Precondiciones	El usuario se encuentra conectado al sistema.
Postcondiciones	Se muestran los datos personales del usuario.

Tabla 3-21: Caso de uso – Consultar y modificar info

Realizar una transferencia	
Actores	Usuario registrado
Descripción	
El usuario realiza una transferencia de fondos desde su cuenta personal a otra cuenta, la cual identifica a través del número de cuenta de la misma.	
Precondiciones	El usuario se encuentra conectado al sistema.
Postcondiciones	Se muestra el código QR necesario para confirmar la transferencia.

Tabla 3-22: Caso de uso - Realizar una transferencia

Volver al menú	
Actores	Usuario registrado
Descripción	
El usuario regresa a la pantalla de menú principal de la aplicación.	
Precondiciones	El usuario se encuentra en una pantalla diferente al menú principal.
Postcondiciones	El usuario se encuentra en el menú principal de la aplicación.

Tabla 3-23: Caso de uso - Volver al menú

Hacer logout	
Actores	Usuario registrado
Descripción	
El usuario hace logout para desconectarse del sistema.	
Precondiciones	El usuario se encuentra conectado al sistema.
Postcondiciones	El usuario se encuentra desconectado del sistema.

Tabla 3-24: Caso de uso - Hacer logout

Hacer login	
Actores	Administrador
Descripción	
El administrador introduce el nombre de usuario y contraseña para acceder al panel de administrador del sistema.	
Precondiciones	El administrador se encuentra en su equipo físico de trabajo.
Postcondiciones	El administrador accede al panel de administración de la aplicación.

Tabla 3-25: Caso de uso - Hacer login (Admin)

Consultar info usuario	
Actores	Administrador
Descripción	
El administrador accede a la información personal de un usuario concreto, pudiendo consultar los datos con los que se dio de alta en el sistema.	
Precondiciones	El administrador se encuentra conectado al sistema.
Postcondiciones	Se muestran los datos del usuario consultado.

Tabla 3-26: Caso de uso - Consultar info usuario

Modificar info usuario	
Actores	Administrador
Descripción	
El administrador modifica la información personal de un usuario, quedando así almacenada en la base de datos del sistema.	
Precondiciones	El administrador se encuentra conectado al sistema.
Postcondiciones	Se almacenan las modificaciones realizadas por el administrador.

Tabla 3-27: Caso de uso - Modificar info usuario

Consultar transferencias	
Actores	Administrador
Descripción	
El administrador consulta un listado con todas las transferencias realizadas por los usuarios en el sistema.	
Precondiciones	El administrador se encuentra conectado al sistema.
Postcondiciones	Se muestra un listado con todas las transferencias registradas.

Tabla 3-28: Caso de uso - Consultar transferencias

Crear usuario	
Actores	Administrador
Descripción	
El administrador da de alta un nuevo usuario introduciendo todos los datos personales necesarios para ello.	
Precondiciones	El administrador se encuentra conectado al sistema.
Postcondiciones	El usuario creado por el administrador queda almacenado en la BBDD.

Tabla 3-29: Caso de uso - Crear usuario

Hacer logout	
Actores	Administrador
Descripción	
El administrador hace logout para desconectarse del sistema.	
Precondiciones	El administrador se encuentra conectado al sistema
Postcondiciones	El administrador se encuentra desconectado del sistema.

Tabla 3-30: Caso de uso - Hacer logout (Admin)

3.5.2 Aplicación móvil

Acciones a realizar por el usuario sobre la aplicación para Android.

Confirmar registro	
Actores	Usuario no registrado
Descripción	
El usuario accede a la aplicación móvil con los datos de acceso introducidos en la aplicación web, completando el proceso de registro en el sistema.	
Precondiciones	El usuario ha cumplimentado el registro en la aplicación web.
Postcondiciones	El usuario se encuentra dado de alta en el sistema.

Tabla 3-31: Caso de uso - Confirmar registro

Realizar captura de QR	
Actores	Usuario registrado
Descripción	
El usuario realiza la captura de un código QR mostrado en la aplicación web haciendo uso de la cámara de su dispositivo móvil.	
Precondiciones	El usuario ha solicitado la realización de una transferencia en la web.
Postcondiciones	Se confirma o deniega la transferencia de dinero.

Tabla 3-32: Caso de uso - Realizar captura de QR

Volver al menú	
Actores	Usuario registrado
Descripción	
El usuario vuelve al menú principal de la aplicación móvil.	
Precondiciones	El usuario se encuentra en una pantalla diferente al menú principal.
Postcondiciones	El usuario se encuentra en el menú principal de la aplicación.

Tabla 3-33: Caso de uso - Volver al menú

3.6 Requisitos de software

La lista de requisitos de software se obtiene a partir de los requisitos de usuario. Se trata de una versión refinada en la que el equipo de desarrollo define, con un lenguaje técnico, el contenido de los requisitos de usuario, de manera que pueda ser utilizado posteriormente para el desarrollo de la aplicación. A su vez se pueden añadir nuevos requisitos que no hayan sido definidos anteriormente pero que el equipo de desarrollo pueda considerar como necesarios para garantizar el correcto funcionamiento de la aplicación.

La definición de los requisitos de software sigue la misma estructura que la de los requisitos de usuario.

3.6.1 Requisitos funcionales

Determinan las acciones que el sistema va a permitir realizar a los usuarios.

RSRF01 – Alta de usuarios			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario darse de alta en el sistema.			

Tabla 3-34: Requisito Funcional – Alta de usuarios

RSRF02 - Login			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario hacer login en el sistema con sus datos de acceso.			

Tabla 3-35: Requisito Funcional – Login

RSRF03 – Consulta movimientos			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario consultar los movimientos y transferencias realizados con su cuenta.			

Tabla 3-36: Requisito Funcional - Consulta movimientos

RSRF04 – Consulta datos			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario consultar los datos personales con los que se ha registrado en el sistema.			

Tabla 3-37: Requisito Funcional - Consulta datos

RSRF05 – Modifica datos			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario modificar los datos personales almacenados en la base de datos del sistema.			

Tabla 3-38: Requisito Funcional - Modifica datos

RSRF06 – Realiza transferencia			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario realizar una transferencia de dinero a otra cuenta.			

Tabla 3-39: Requisito Funcional - Realiza transferencia

RSRF07 – Confirma transferencia			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario confirmar una transferencia haciendo uso de la aplicación móvil.			

Tabla 3-40: Requisito Funcional - Confirma transferencia

RSRF08 – Volver al menú			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario volver al menú principal en cualquier momento.			

Tabla 3-41: Requisito Funcional - Volver al menú

RSRF09 - Logout			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un usuario desconectarse del sistema.			

Tabla 3-42: Requisito Funcional - Logout

RSRF10 – Consulta información			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un administrador consultar los datos personales de cualquier usuario.			

Tabla 3-43: Requisito Funcional - Consulta información

RSRF11 – Modifica información			
Prioridad	☐ Alta ☒ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
El sistema permitirá a un administrador modificar los datos personales de un usuario concreto.			

Tabla 3-44: Requisito Funcional - Modifica información

RSRF12 – Consulta movimientos			
Prioridad	☐ Alta ☐ Media ☒ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☐ Alta ☒ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un administrador consultar el listado completo de transferencias realizadas por los usuarios en el sistema.			

Tabla 3-45: Requisito Funcional - Consulta movimientos

RSRF13 – Nuevo usuario			
Prioridad	☐ Alta ☐ Media ☒ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
El sistema permitirá a un administrador dar de alta nuevos usuarios.			

Tabla 3-46: Requisito Funcional - Nuevo usuario

3.6.2 Requisitos no funcionales

Determinan las restricciones y condiciones especiales que debe cumplir el sistema.

RSRN01 – Identificador del terminal			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☐ Alta ☒ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
En la base de datos se almacena un identificador único del terminal móvil para cada usuario, que será utilizado para generar los códigos QR para confirmar las transferencias.			

Tabla 3-47: Requisito No Funcional - Identificador del terminal

RSRN02 – Usuario no registrado			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☐ Alta ☒ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Cuando el usuario elige su nombre de usuario, se comprueba en la base de datos que dicho nombre no está siendo usado por otro usuario.			

Tabla 3-48: Requisito No Funcional – Usuario no registrado

RSRN03 – Hash de contraseña			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
En la base de datos se almacena el Hash de la contraseña introducida por el usuario, y se compara dicho Hash con el introducido por el usuario cada vez que hace login.			

Tabla 3-49: Requisito No Funcional - Hash en contraseña

RSRN04 – Información del servidor cifrada			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
El documento donde se almacena la información y claves del servidor se cifra con un algoritmo AES.			

Tabla 3-50: Requisito No Funcional – Información del servidor cifrada

RSRN05 – Campos en blanco			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☒ Alta ☐ Media ☐ Baja
Descripción			
Se comprueba que cuando un usuario rellena el formulario de registro de la aplicación, no se dejen campos sin completar.			

Tabla 3-51: Requisito No Funcional – Campos en blanco

RSRN06 – Caracteres extraños			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
Se comprueba que cuando un usuario rellena el formulario de registro de la aplicación, no se introducen caracteres que puedan poner en peligro la integridad de la base de datos.			

Tabla 3-52: Requisito No Funcional – Caracteres extraños

RSRN07 – Ataques web			
Prioridad	☒ Alta ☐ Media ☐ Baja	Necesidad	☒ Esencial ☐ Deseable
Claridad	☒ Alta ☐ Media ☐ Baja	Verificabilidad	☐ Alta ☐ Media ☒ Baja
Descripción			
Se siguen los procedimientos de protección necesarios para evitar la posibilidad de atacar el sistema mediante vulnerabilidades web.			

Tabla 3-53: Requisito No Funcional - Ataques web

RSRN08 – Coherencia de diseño			
Prioridad	☐ Alta ☐ Media ☒ Baja	Necesidad	☐ Esencial ☒ Deseable
Claridad	☐ Alta ☒ Media ☐ Baja	Verificabilidad	☐ Alta ☒ Media ☐ Baja
Descripción			
La interfaz de ambas aplicaciones guarda coherencia y mantiene un patrón de color y diseño uniforme y agradable.			

Tabla 3-54: Requisito No Funcional - Coherencia de diseño

3.7 Matriz de trazabilidad

La matriz de trazabilidad se utiliza para comprobar que cada uno de los requisitos de usuario ha quedado representado por al menos un requisito de software, de manera que se asegura que no hay ninguna solicitud original que se haya pasado por alto.

		Requisitos capacidad												Requisitos restricción				
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5
Requisitos funcionales	1	x																
	2		x															
	3			x														
	4				x													
	5				x													
	6					x												
	7						x											
	8							x										
	9								x									
	10									x								
	11										x							
	12											x						
	13												x					
Requisitos no funcioanles	1													x				
	2																	
	3														x			
	4															x		
	5																	
	6																x	
	7																x	
	8																	x

Tabla 3-55: Matriz de trazabilidad

Tal y como se puede comprobar, cada uno de los requisitos de usuario se corresponde con al menos un requisito de software, de manera que se puede asegurar que no se ha pasado por alto ninguna solicitud.

3.8 Interfaz de usuario

La aplicación consiste en un sistema bancario que permite a los usuarios consultar sus fondos y movimientos y realizar transferencias. Es necesario establecer una serie de principios que garanticen que el diseño de la interfaz sea sencillo de utilizar por los usuarios y les ayude a cumplir sus necesidades.

3.8.1 Principios generales de la interfaz

Para cumplir con el objetivo anterior, se determina que la interfaz de la aplicación deberá ser:

- **Fácil de utilizar**: Se deberán tener en cuenta tanto las necesidades del usuario final como su experiencia. El usuario no debe verse, en ningún caso, forzado a adaptarse a la interfaz, de manera que ésta debe emplear sistemas conocidos como iconos o menús que resulten familiares al usuario, garantizando así un uso de la aplicación más amigable.
- **Intuitiva**: La navegación que se realice en la aplicación debe seguir un patrón basado en la lógica, que permita a los usuarios cumplir sus requisitos de la manera más sencilla y evitando que deban pensar cómo realizar cada acción.
- **Informativa**: La aplicación debe mantener en todo caso informado al usuario. Éste debe conocer en todo momento el menú en que se encuentra y debe poder volver al menú principal en cualquier momento. A su vez, en caso de darse condiciones especiales o alertas, se informará al usuario con una notificación, permitiéndole realizar alguna acción asociada en caso de que sea necesario.
- **Consistente**: Los componentes similares de la aplicación guardarán un patrón y esquema gráfico similar, de manera que se garantice un grado de consistencia alto, asegurando la mayor comodidad para el usuario.

3.8.2 Comportamiento dinámico de la interfaz

A partir del catálogo de usuarios que va a interactuar con el sistema se determina el mapa de navegación de la aplicación. Dicho mapa de navegación se encuentra estrechamente relacionado con los casos de uso a realizar por los usuarios del sistema.

El sistema dispone de dos tipos de usuarios que harán uso del mismo. A su vez, uno de ellos desempeñará dos posibles roles dependiendo del momento en que haga uso de la aplicación:

- **Administrador**: Tiene acceso completo y sin restricciones sobre el sistema. Puede realizar acciones únicamente reservadas a su rol.

- **Usuario común**: Tiene acceso a todas las acciones y consultas sobre su propia cuenta privada. Puede desempeñar dos roles distintos:
  - **Usuario no registrado**: Es la primera vez que accede al sistema. Tiene que pasar por el proceso de registro para que sus datos queden almacenados en la base de datos.
  - **Usuario registrado**: Tiene acceso sobre el sistema. Al hacer login accede al menú principal de la aplicación desde donde tiene acceso a todas las acciones que se le ofrecen.

El comportamiento de la interfaz para cada uno de los usuarios es el siguiente:

- Usuario común:

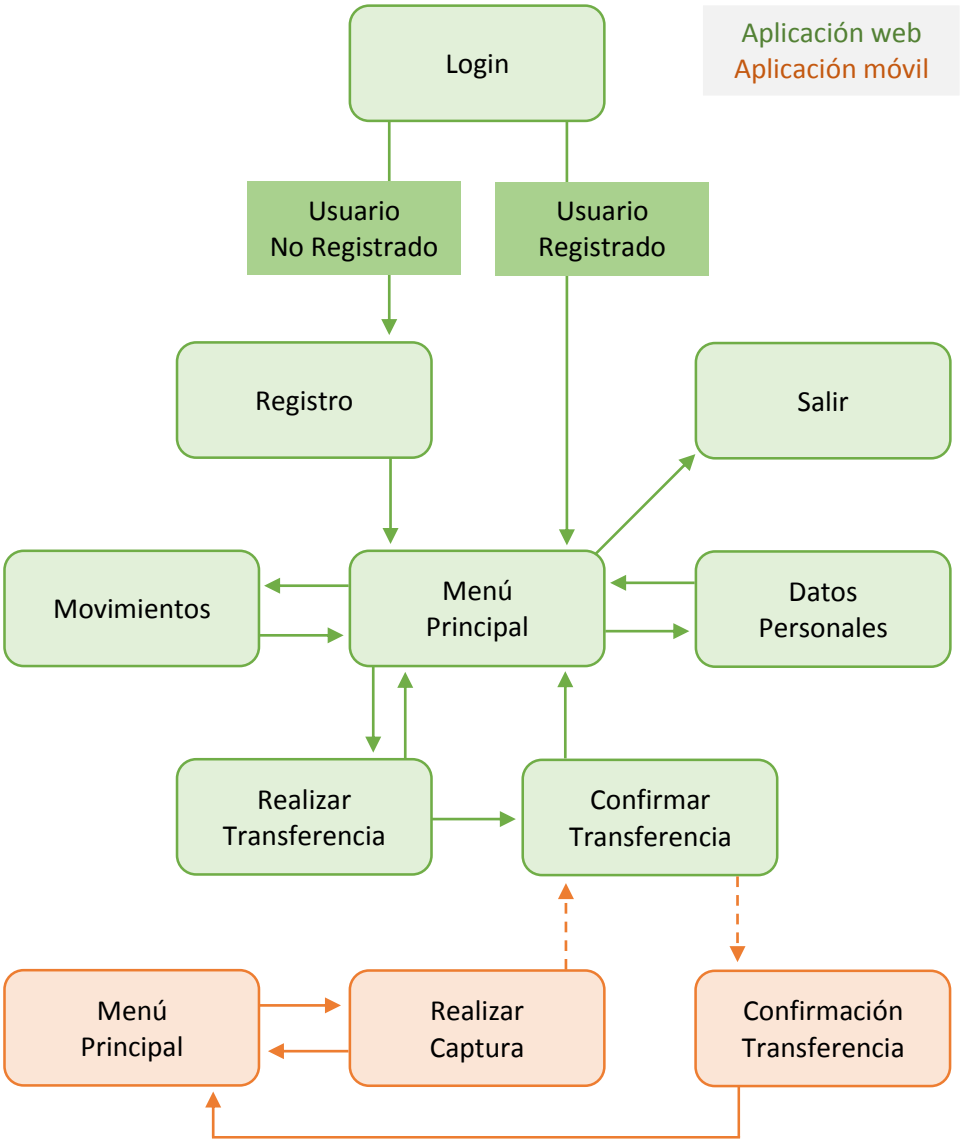


Figura 3-3: Comportamiento dinámico: Mapa de navegación del usuario común

- Administrador:

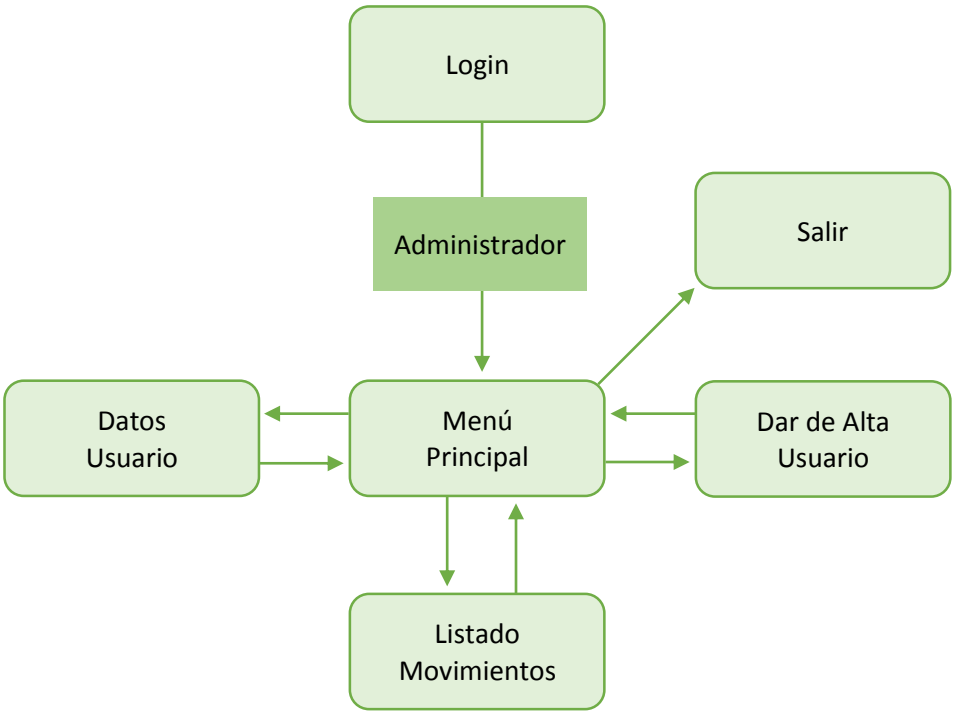


Figura 3-4: Comportamiento dinámico: Mapa de navegación del administrador

3.8.3 Especificación de los formatos de la interfaz de pantalla

La aplicación se compone de dos módulos diferentes, cada uno de los cuales dispone de sus propios formatos de interfaz. Los módulos de que se compone el sistema son los siguientes:

- **Aplicación web**: Define todo el entorno web que da forma al sistema. Ofrece la mayor parte de la funcionalidad del mismo. Se conforma a partir de dos estructuras o plantillas:
  - **Vista navegación**: Se caracteriza por la utilización de un fondo de tonos claros con menús y botones en tonos oscuros, y utilizando detalles en color naranja. Este tipo de vista favorece la visualización de información, formularios y demás componentes que requieran la atención del usuario. A su vez, desde cada una de las páginas que implementan esta plantilla, se ofrece acceso instantáneo a todas las demás. Es utilizada por las siguientes vistas:
    - Menú principal
    - Últimos movimientos
    - Nueva transferencia
    - Información personal

- **Vista formulario**: Se caracteriza por la utilización de un fondo naranja y cuadros de texto blancos a rellenar por el usuario. Es utilizado en las pantallas que requieren que el usuario introduzca datos antes de hacer login en el sistema. Es utilizada por las siguientes vistas:
    - Login
    - Registro

- **Aplicación móvil**: Consiste en el subsistema que se instala en el terminal Android y se utiliza para validar las transferencias realizadas por el usuario. Se conforma a partir de la siguiente plantilla:
    - **Vista móvil**: Se caracteriza por su simplicidad, basando su diseño en la tendencia *flat*, seguida por gran parte de las aplicaciones actuales. Se conforma por un fondo oscuro similar al utilizado en la aplicación web, y un logo de gran tamaño en la parte superior de la aplicación. De la misma forma, los botones poseen un gran tamaño y utilizan tonos naranja y blanco, al igual que los textos, siguiendo el patrón de color definido en el apartado web. Este patrón es utilizado en las siguientes vistas:
        - Main
        - Registro
        - Menú principal
        - Captura
        - Resultado

Diseño

El objetivo del diseño del sistema es la definición de la arquitectura del sistema y el entorno tecnológico que le va a dar soporte, así como especificar, de manera detallada, los componentes del sistema de información. A partir de los resultados obtenidos, se generan las especificaciones de construcción del sistema, así como los requisitos de implantación.

4.1 Arquitectura del sistema

Para poder llevar a cabo la implementación de la aplicación se debe detallar la arquitectura software que se va a emplear. El control del sistema será determinado por el patrón *Modelo Vista Controlador* (MVC), separando los distintos componentes del sistema para facilitar su manejo.

Modelo Vista Controlador es un patrón de desarrollo software compuesto por tres niveles, permitiendo así separar los datos, la interfaz de usuario y la lógica interna de la aplicación.

A continuación se detalla cada uno de los elementos de este esquema:

- **Modelo**: Consiste en la representación de la información en el sistema. Su utilización de manera conjunta con la Vista permite mostrar la información al usuario, y es accedido por el Controlador para consultar o modificar los datos.
- **Vista**: Da forma al Modelo para permitir al usuario la interacción con éste. Generalmente se corresponde con la interfaz de usuario.
- **Controlador**: Obtiene, trata y responde a los eventos recibidos por el sistema, generalmente solicitados por el usuario o la propia aplicación. Interactúa de manera conjunta con los otros dos componentes del patrón.

El esquema de funcionamiento del patrón Modelo Vista Controlador es el siguiente:

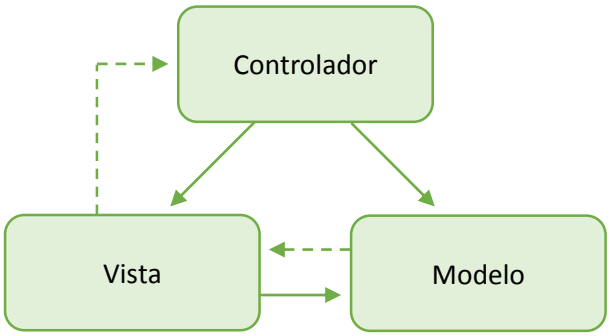


Figura 4-1: Modelo Vista Controlador

Para el sistema actual el modelo se aplica de la siguiente manera:

- **Modelo**: Se corresponde con la base de datos donde se almacena la información del sistema y los métodos que permiten su manejo.
- **Vista**: Está compuesta por cada uno de los layouts que determinan la interfaz gráfica de la aplicación. Recibe del Modelo la información a mostrar.
- **Controlador**: Se trata de la mayor parte de la aplicación Java que compone el sistema. Se encarga de manejar la información que posee el sistema, y su utilización conjunta con la vista permite mostrarla al usuario.

4.2 Subsistemas de diseño

El sistema general que compone la aplicación se divide en subsistemas para facilitar tanto su estudio como su posterior implementación, de manera que se permite el desarrollo de distintos módulos de forma paralela, acelerando el proceso de desarrollo de la aplicación y permitiendo la prueba de los distintos componentes del sistema de manera independiente y diferenciada.

4.2.1 Aplicación web

A continuación se detallan los subsistemas que componen y dan forma al apartado web del sistema.

4.2.1.1 Subsistema de registro

El objetivo del subsistema de registro es el de dar de alta los datos personales del usuario cuando se accede al sistema por primera vez y se procede a registrarse en el mismo. Dicho proceso solamente es necesario realizarlo una vez, y una vez finalizada la edición de los datos, el subsistema se comunica con la base de datos para realizar el almacenamiento de la información introducida.

4.2.1.2 Subsistema de autenticación

El subsistema de autenticación es el encargado de activar el uso del perfil del usuario cuando éste hace login en el sistema con los datos con los que se ha registrado en el mismo. Se hace uso de una sesión para controlar el estado del usuario en el sistema.

4.2.1.3 Subsistema de acceso a la aplicación

El subsistema de acceso a la aplicación se encarga de que se realicen todas las acciones de acceso a datos necesarias para la correcta ejecución del sistema. Para ello se realizan las consultas necesarias sobre la base de datos haciendo uso de la sesión habilitada en el proceso de login. De esta forma, cuando el usuario obtiene acceso a todas las funcionalidades de la aplicación, los datos necesarios ya se encuentran cargados, evitando detenciones o errores en el proceso de ejecución.

4.2.1.4 Subsistema de movimientos

El subsistema de movimientos realiza una consulta sobre todas las transferencias realizadas y recibidas por el usuario y las muestra en una tabla dinámica en el apartado web correspondiente. Dicha lista viene acompañada de todos los datos de cada una de las transferencias.

4.2.1.5 Subsistema de transferencias

El subsistema de transferencias permite a un usuario realizar una transferencia a otra cuenta y da la posibilidad de confirmar dicha transferencia haciendo uso de la aplicación móvil. Se subdivide en los siguientes subsistemas:

4.2.1.5.1 Subsistema de formulario de transferencia

Dispone de un formulario donde el usuario debe introducir el número de cuenta sobre el que desea realizar la transferencia, la cantidad de dinero a mover, y se le permite rellenar un campo opcional de concepto.

4.2.1.5.2 Subsistema de confirmación de transferencia

A partir de la información proporcionada por el usuario y otra producida por el sistema, se genera un código QR único que permite validar la transferencia haciendo uso de la aplicación móvil.

4.2.1.6 Subsistema de datos personales

El subsistema de datos personales permite al usuario consultar los datos personales con los que se registró en el sistema, y modificar dichos datos. Para ello se hace uso de un formulario similar al utilizado en el procedimiento de registro.

4.2.2 Aplicación móvil

A continuación se detallan los subsistemas que componen y dan forma al apartado móvil del sistema:

4.2.2.1 Subsistema de acceso a la aplicación

El subsistema de acceso a la aplicación se encarga de comprobar los datos registrados en la aplicación y redireccionar al usuario a una u otra vista en función de los resultados obtenidos. Para ello accede tanto a las preferencias de la aplicación como al servicio web que permite la consulta de los datos de la base de datos.

4.2.2.2 Subsistema de registro

El Subsistema de registro permite a un usuario finalizar el proceso de registro iniciado en la aplicación web. Dicho procedimiento se finaliza con la introducción de sus datos de acceso la primera vez que se accede a la aplicación móvil. Una vez finalizado, no se volverá a solicitar la introducción de dichos datos.

4.2.2.3 Subsistema de menú principal

El subsistema de menú principal permite a un usuario comenzar el procedimiento de validación de una transferencia.

4.2.2.4 Subsistema de captura

El subsistema de captura hace uso de la cámara del dispositivo para buscar un código QR en las imágenes capturadas en tiempo real. En el momento en que se detecta un código QR, el sistema intenta, de forma automática, su validación.

4.2.2.5 Subsistema de resultado

El subsistema de resultado es el encargado de procesar el contenido del código QR, descifrar la información almacenada en el mismo, devolver el resultado al servidor y recibir la respuesta de éste.

4.2.2.6 Subsistema de confirmación

El subsistema de confirmación permite conocer al usuario el resultado obtenido en el procedimiento de confirmación de la transferencia. Muestra *transferencia confirmada* si el servidor la ha dado por buena, o *transferencia rechazada* en caso contrario.

4.3 Especificación del entorno tecnológico

Para realizar una especificación del entorno tecnológico se determinan los requisitos básicos que deben cumplir los equipos en los que se va a utilizar la aplicación.

La aplicación web deberá ejecutarse en un equipo con acceso a internet. Se recomienda la utilización de un equipo con un mínimo de 500MHz de procesador y 256MB de RAM, y la resolución del monitor no debe ser inferior a 1024x768 para la correcta visualización de la página web. Se ha comprobado su correcto funcionamiento en los exploradores Internet Explorer, Mozilla Firefox y Google Chrome.

La instalación de la aplicación móvil se debe realizar sobre un Smartphone. Dicho dispositivo deberá disponer de, al menos, un procesador de 600MHz y 10MB de espacio disponible para realizar la instalación. A su vez la pantalla no deberá tener una resolución inferior a 240x320 para permitir la correcta visualización de la aplicación. El sistema operativo del dispositivo móvil deberá ser Android, y la versión del mismo no deberá ser inferior a la v4.0 Ice Cream Sandwich.

4.4 Diseño de clases

El diseño de clases describe la estructura del sistema a partir de la lista de clases que lo componen. Se definen tanto las clases como los atributos que poseen y sus relaciones.

4.4.1 Identificación de atributos y operaciones

Se detallan a continuación las distintas clases que componen el sistema, acompañadas de una breve descripción y los atributos y operaciones de la misma.

Usuario	
Descripción	
Clase que contiene los datos personales de un usuario de la aplicación.	
Atributos	username password nombre apellidos dni sexo fecha_nacimiento dirección teléfono email identificador_tlf número_cuenta ccv admin
Operaciones	

Tabla 4-1: Clase Usuario

Transferencia	
Descripción	
Clase que contiene la información relativa a una transferencia.	
Atributos	id username cuenta_origen cuenta_destino cantidad fecha concepto confirmada
Operaciones	

Tabla 4-2: Clase Transferencia

Fondos	
Descripción	
Clase donde se asocia una cantidad de fondos a un usuario.	
Atributos	username saldo
Operaciones	

Tabla 4-3: Clase Fondos

Claves	
Descripción	
Clase donde se registra la clave pública del usuario para poder cifrar los mensajes destinados a éste.	
Atributos	username clave
Operaciones	

Tabla 4-4: Clase Claves

Vista_login	
Descripción	
Vista conformada por el formulario de acceso al sistema. Permite a un usuario ingresar su nombre de usuario y contraseña.	
Atributos	username password
Operaciones	Enviar()

Tabla 4-5: Clase Vista_login

Vista_registro	
Descripción	
Vista conformada por el formulario de registro en el sistema. Permite al usuario ingresar sus datos personales para darse de alta.	
Atributos	username password nombre apellidos dni sexo fecha_nacimiento dirección teléfono email
Operaciones	Enviar() Volver()

Tabla 4-6: Clase Vista_registro

Vista_base	
Descripción	
Vista utilizada como base para todas las vistas del sistema una vez se ha hecho login. Dichas vistas heredan de ésta atributos y métodos. Ofrece acceso a toda la funcionalidad del sistema.	
Atributos	username
Operaciones	Últimos_movimientos() Nueva_transferencia() Información_personal() Salir()

Tabla 4-7: Clase Vista_base

Vista_menú_principal	
Descripción	
Vista a la que accede el usuario cuando hace login.	
Atributos	nombre
Operaciones	

Tabla 4-8: Clase Vista_menú_principal

Vista_últimos_movimientos	
Descripción	
Vista que ofrece un listado de los movimientos de fondos entrantes y salientes realizados sobre la cuenta del usuario.	
Atributos	cuenta_origen cuenta_destino cantidad fecha concepto confirmada
Operaciones	

Tabla 4-9: Clase Vista_últimos_movimientos

Vista_nueva_transferencia	
Descripción	
Vista que ofrece al usuario un formulario donde rellenar los datos para realizar una nueva transferencia.	
Atributos	cuenta_destino cantidad concepto
Operaciones	Enviar()

Tabla 4-10: Clase VIsta_nueva_transferencia

Vista_confirma_transferencia	
Descripción	
Vista que muestra un código QR con la información necesaria para confirmar la transferencia haciendo uso de la aplicación para Android.	
Atributos	
Operaciones	

Tabla 4-11: Clase Vista_confirma_transferencia

Vista_información_personal	
Descripción	
Vista que muestra los datos con los que el usuario se ha dado de alta en el sistema. Permite, a su vez, la modificación de dichos datos.	
Atributos	nombre apellidos dni sexo fecha_nacimiento dirección teléfono email
Operaciones	Aplicar()

Tabla 4-12: Clase Vista_información_personal

Vista_móvil_base	
Descripción	
Vista de la aplicación móvil utilizada como base para todas las vistas de dicha aplicación. Define el formato y la estructura de dichas vistas.	
Atributos	
Operaciones	

Tabla 4-13: Clase Vista_móvil_base

Vista_móvil_registro	
Descripción	
Vista de la aplicación móvil que solicita al usuario la introducción de su nombre de usuario y su contraseña para completar el registro.	
Atributos	username password
Operaciones	Enviar()

Tabla 4-14: Clase Vista_móvil_registro

Vista_móvil_menú_principal	
Descripción	
Vista de la aplicación móvil cuando se accede a la aplicación una vez completado el registro.	
Atributos	
Operaciones	Realizar_captura()

Tabla 4-15: Clase Vista_móvil_menú_principal

Vista_móvil_captura	
Descripción	
Vista de la aplicación móvil que, haciendo uso de la cámara del dispositivo, busca un código QR.	
Atributos	código_codificado
Operaciones	Descifrar() Volver()

Tabla 4-16: Clase Vista_móvil_captura

Vista_móvil_resultado	
Descripción	
Vista de la aplicación móvil que confirma al usuario si la operación ha sido aceptada o rechazada.	
Atributos	
Operaciones	Volver()

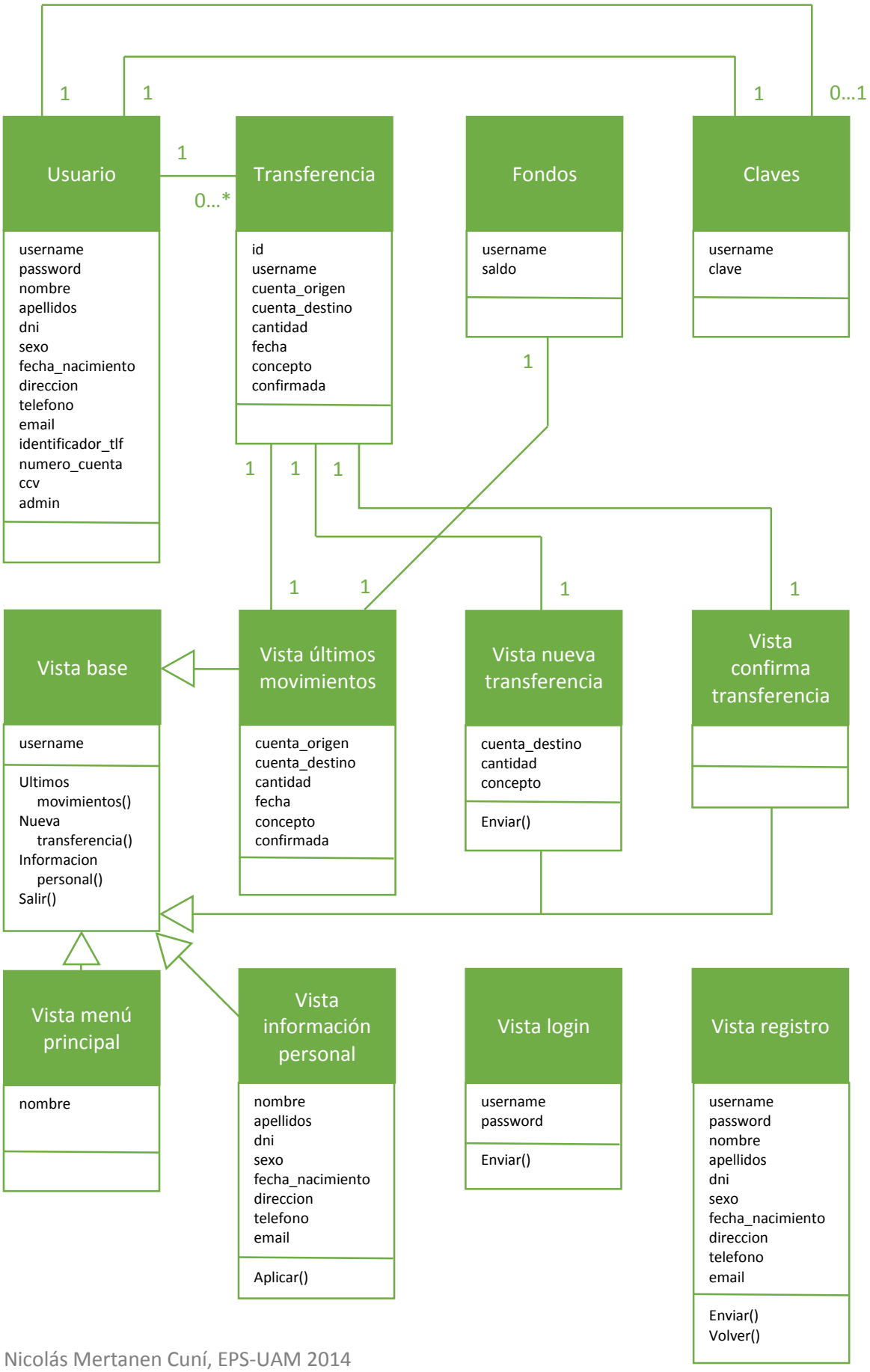
Tabla 4-17: Clase Vista_móvil_resultado

Manejador_BBDD	
Descripción	
Clase que se encarga de realizar las operaciones solicitadas sobre la base de datos.	
Atributos	
Operaciones	

Tabla 4-18: Clase Manejador_BBDD

4.4.2 Modelo de clases

El modelo de clases resultante que determina el funcionamiento del sistema es el siguiente:



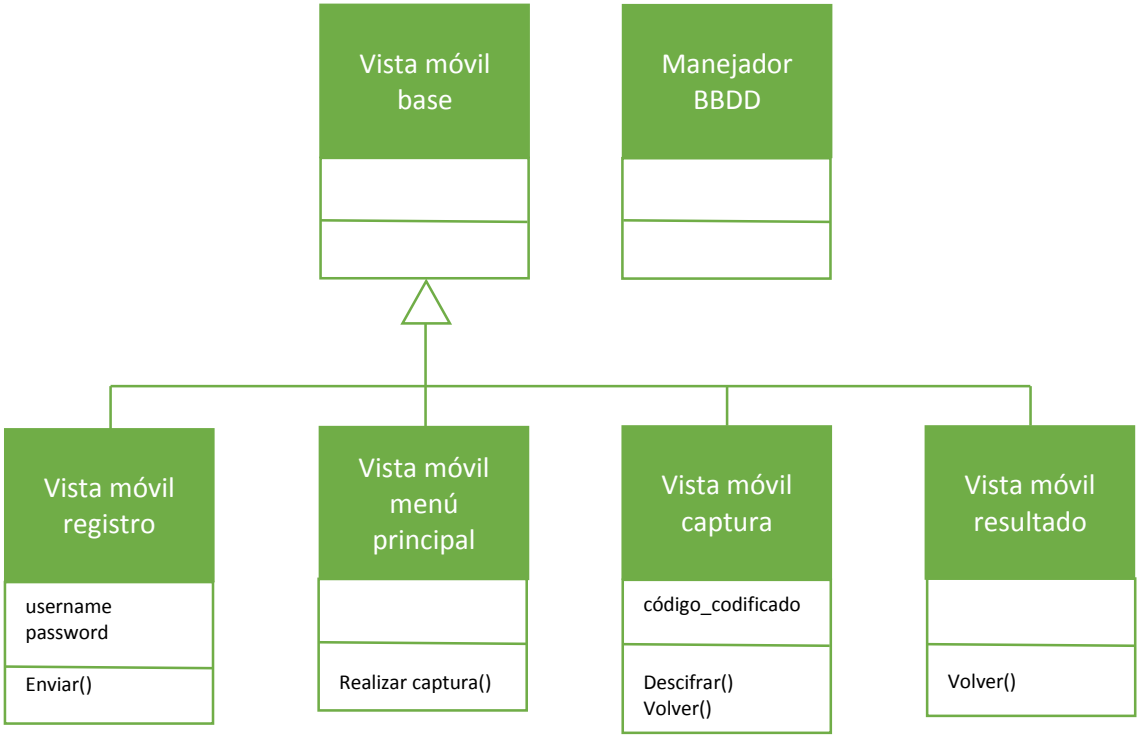


Figura 4-2: Modelo de Clases

4.4.3 Clases asociadas a casos de uso

A continuación se identifican las clases necesarias para llevar a cabo cada uno de los casos de uso. Dichas clases serán catalogadas en función del uso que se haga de las mismas en el sistema.

Clases de entidad

- **Usuario**: Clase que contiene los datos personales y datos de acceso de un usuario de la aplicación.
- **Transferencia**: Clase que relaciona a un usuario con una transferencia realizada. Contiene los datos de dicha transferencia.
- **Fondos**: Clase que relaciona a un usuario con el saldo de su cuenta personal.
- **Claves**: Clase que relaciona a un usuario con su clave pública. Utilizada para que el servidor pueda cifrar los mensajes que le envía.

Clases de interfaz de usuario

- **Vista_registro**: Clase que contiene el diseño de la vista de registro de la aplicación. Sobre dicha vista, el usuario introduce sus datos personales para darse de alta en el sistema.
- **Vista_login**: Clase que contiene el diseño de la vista de login. Dicha vista solicita introducir un usuario y una contraseña para acceder al sistema.

- **Vista_base**: Clase que contiene el diseño de la vista base. Dicha vista se utiliza como estructura para todas las vistas de la aplicación interna. Además, todas las clases que heredan de esta vista, heredan los métodos y acciones que implementa.
- **Vista_menú_principal**: Clase que contiene el diseño de la vista del menú principal. Presenta una bienvenida al usuario y ofrece acceso a toda la funcionalidad del sistema.
- **Vista_últimos_movimientos**: Clase que contiene el diseño de la vista de últimos movimientos. Dicha vista ofrece una relación de los movimientos de fondos realizados con la cuenta del usuario.
- **Vista_nueva_transferencia**: Clase que contiene el diseño de la vista de nueva transferencia. Dicha vista permite realizar una transferencia de dinero de la cuenta del usuario a otra cuenta.
- **Vista_confirma_transferencia**: Clase que contiene el diseño de la vista de la confirmación de una transferencia. Presenta un código QR que el usuario debe escanear con su terminal móvil para confirmar la transferencia de dinero.
- **Vista_información_personal**: Clase que contiene el diseño de la vista de información personal. Dicha vista ofrece la información con que el usuario se dio de alta en el sistema, y le ofrece la posibilidad de modificar dicha información.
- **Vista_móvil_base**: Clase que contiene el diseño de la vista base de la aplicación móvil. Dicha estructura se centra en el diseño de la aplicación móvil, que será heredado por todas las demás vistas.
- **Vista_móvil_registro**: Clase que contiene el diseño de la vista de registro de la aplicación móvil. Solicita el nombre de usuario y la contraseña del usuario cuando éste ya ha formalizado el registro en la aplicación web.
- **Vista_móvil_menú_principal**: Clase que contiene el diseño de la vista del menú principal de la aplicación móvil. Ofrece acceso a la captura de códigos QR.
- **Vista_móvil_captura**: Clase que contiene el diseño de la vista de captura de la aplicación móvil. Hace uso de la cámara del dispositivo para capturar un código QR.
- **Vista_móvil_resultado**: Clase que contiene el diseño de la vista de resultado de la aplicación móvil. Muestra al usuario el resultado de la confirmación de la transferencia realizado al capturar el código QR.

Clases de control

- **Manejador_BBDD**: Clase utilizada como intérprete entre la aplicación y la base de datos online. Todas las operaciones que se realicen sobre la base de datos deberán utilizar una instancia de esta clase.

Se determinan las clases de diseño y subsistemas cuyas instancias se emplean para llevar a cabo cada uno de los casos de uso.

Caso de uso	Clase asociada
Darse de alta en la aplicación	➢ Usuario ➢ Fondos ➢ Claves ➢ Vista_registro ➢ Manejador_BBDD
Hacer login	➢ Usuario ➢ Vista_login ➢ Manejador_BBDD
Consultar movimientos	➢ Usuario ➢ Transferencia ➢ Fondos ➢ Vista_base ➢ Vista_movimientos ➢ Manejador_BBDD
Realizar una transferencia	➢ Usuario ➢ Transferencia ➢ Fondos ➢ Vista_base ➢ Vista_nueva_transferencia ➢ Vista_confirma_transferencia ➢ Manejador_BBDD
Consultar y modificar info	➢ Usuario ➢ Vista_base ➢ Vista_información_personal ➢ Manejador_BBDD
Volver al menú	➢ Usuario ➢ Vista_base ➢ Vista_menú_principal ➢ Manejador_BBDD
Hacer logout	➢ Usuario ➢ Vista_login ➢ Manejador_BBDD
Confirmar registro	➢ Usuario ➢ Vista_móvil_registro ➢ Manejador_BBDD
Realizar captura de QR	➢ Usuario ➢ Transferencia ➢ Vista_móvil_captura ➢ Vista_móvil_resultado ➢ Manejador_BBDD
Volver al menú	➢ Vista_móvil_menú_principal

Tabla 4-19: Clases asociadas a casos de uso

4.5 Modelo físico de datos

El modelo físico de datos se utiliza para determinar la forma en que se guarda la información en la base de datos de la aplicación.

4.5.1 Base de datos de información de usuarios

Se realiza un estudio del subsistema donde se almacena la información de los usuarios para detectar cuáles son los puntos más vulnerables del mismo. El subsistema está compuesto por cuatro tablas donde se almacenará la información online necesaria para la interacción entre los distintos componentes de la aplicación. Dichas tablas son las siguientes:

- **Usuarios**: Contiene la información de todos los usuarios de la aplicación. Por un lado se almacena la información personal, introducida en el momento de registro en el sistema. Por otro lado, se almacena la información relativa a la cuenta bancaria con que se identifica el usuario en el sistema. Por último, se registra el identificador único del terminal móvil, utilizado para validar las transferencias.
- **Transferencias**: Utilizada para relacionar a dos usuarios a partir de una transferencia de dinero realizada de uno a otro. El identificador de la transferencia es un valor aleatorio que identifica de manera única a la transferencia y se utiliza en el proceso de validación. El campo *confirmada* se utiliza para llevar un control más rápido de transferencias validadas.
- **Fondos**: Guarda la relación entre cada uno de los usuarios del sistema y el saldo total de que disponen. El valor de dicho saldo se modifica en el momento en que una transferencia es confirmada.
- **Claves**: Relaciona a cada usuario del sistema con su clave pública. Dicha clave se emplea para que el servidor cifre los mensajes destinados al usuario, de manera que sea éste el único que pueda consultar su contenido.

Para realizar el estudio de manera efectiva se analizan tres posibles factores determinantes de vulnerabilidades:

- **Mayor volumen de datos**: La tabla usuarios dispone de un mayor volumen de datos que las otras dos tablas, ya que se almacena una gran cantidad de información para cada uno de los usuarios del sistema.
- **Mayor cantidad de información en forma de texto plano**: La tabla Transferencias dispone de mayor cantidad de información almacenada que las otras tablas, ya que, a

pesar de almacenar un número menor de campos que la tabla Usuarios, se realiza un alto número de inserciones por usuario.

- **Mayor volumen de accesos y consultas**: La tabla Usuarios presenta el mayor volumen de accesos y consultas del sistema, ya que se utiliza constantemente para validar la identidad del usuario o conocer datos necesarios para realizar operaciones en otras tablas.

A partir de la información anterior, se genera el siguiente modelo físico de datos:

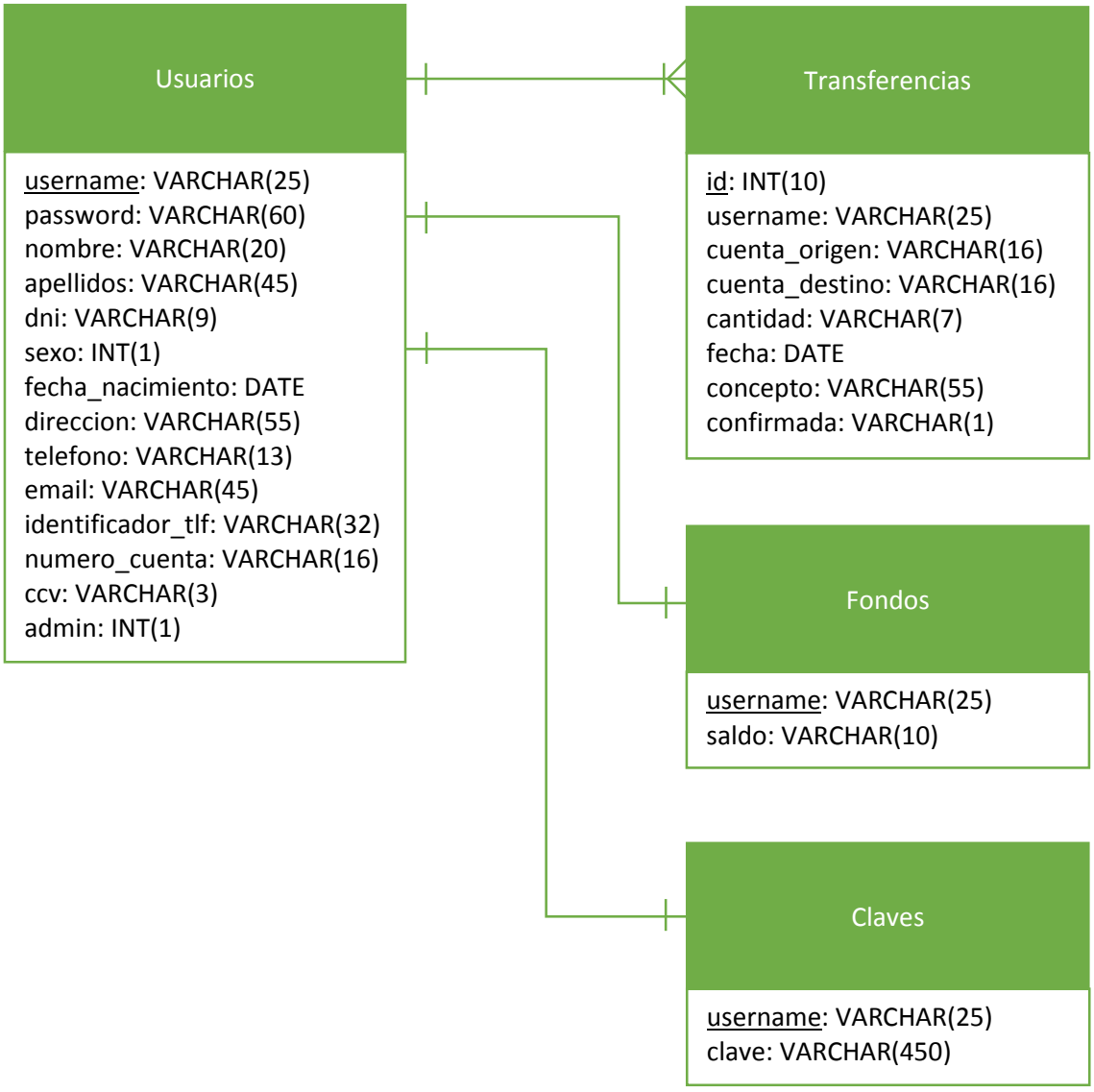


Figura 4-3: Modelo físico de datos de información de usuarios

Implementación

Se realiza un seguimiento del proceso de implementación del sistema. Dicho procedimiento recoge la realización de todas las actividades necesarias para la producción del mismo siguiendo la estrategia de implantación que ya se determinó en el Estudio inicial del sistema. Adicionalmente se definen las pruebas necesarias en función de los requisitos de software, para asegurar que se cumplen las solicitudes originales, y garantizar el correcto funcionamiento de la aplicación.

5.1 Equipo de desarrollo

A continuación se detallan las características del equipo sobre el que se llevará a cabo el desarrollo del sistema.

5.1.1 Hardware

Se detallan las características, tanto del equipo de desarrollo, como de los equipos fijos y los terminales móviles sobre los que se llevarán a cabo las pruebas de la aplicación.

5.1.1.1 Equipo principal de desarrollo

El proyecto se desarrolla, de manera íntegra, en un equipo con las siguientes características, suponiendo esto tanto la codificación de ambas aplicaciones, como el proceso de diseño de las mismas y las primeras pruebas realizadas para comprobar el funcionamiento del entorno web:

Equipo portátil Asus N550JK con procesador Intel Core I7-4700HQ de 2.40GHz, 8GB de memoria DDR3, disco duro principal de 1TB y disco auxiliar de 2TB conectado por USB 3.0, y sistema operativo Windows 8.1 instalado.

5.1.1.2 Equipo físico de pruebas

Además del equipo principal, sobre el que se realizan las primeras pruebas de funcionamiento, se utilizan los siguientes equipos para garantizar el funcionamiento de la aplicación bajo diferentes configuraciones:

Equipo de sobremesa con procesador Intel Core 2 Quad Q6600 de 2.40GHz, 4GB de memoria RAM DDR2, disco duro principal de 250GB y disco secundario de 500GB, y sistema operativo Windows 7 Ultimate instalado.

Equipo portátil Packard Bell Easynote con procesador AMD E1 1200 de 1.40GHz, 8GB de memoria DDR3, disco duro principal de 500GB y sistema operativo Windows 8.1 instalado.

5.1.1.3 Terminal móvil principal para pruebas

El desarrollo de la aplicación móvil se desarrolla en el equipo anteriormente indicado. Dicho programa se compila y se carga en un terminal móvil que se encuentra conectado al equipo anterior. Las características de dicho terminal móvil son las siguientes:

Motorola Moto G

- Procesador: Qualcomm Quad-Core A7 1.2GHz.
- Pantalla: 1280 X 720 4.5”.
- Android: v4.4 KitKat.

5.1.1.4 Terminales móviles adicionales

De la misma forma, se hace uso de una serie de dispositivos para garantizar el correcto funcionamiento y visualización de la aplicación en distintos terminales. La lista de dispositivos en los que se instala la aplicación es la siguiente:

Nexus 5

- Procesador: Snapdragon 800 2.26GHz.
- Pantalla: 1920 X 1080 4.95”.
- Android: v4.4 KitKat.

Samsung S3 Mini

- Procesador: ARM Cortex-A9 Dual-Core 1GHz.
- Pantalla: 800 X 480 4.0”.
- Android: v4.1 Jelly Bean

LG Optimus L7

- Procesador: Qualcomm Snapdragon 1GHz.
- Pantalla: 800 X 480 4.3”.
- Android: v4.0 Ice Cream Sandwich.

5.1.2 Software

Para llevar a cabo el desarrollo del proyecto se hace uso de una serie de aplicaciones que permiten, tanto la codificación, como el diseño y montaje de las mismas. La lista de aplicaciones de que se hace uso es la siguiente:

5.1.2.1 Software de programación

Utilizado para la creación y codificación de las aplicaciones que componen el sistema:

- Eclipse IDE
  - Java EE Developers
  - Android Development Tools
- Notepad++
- XAMPP
  - phpMyAdmin

5.1.2.2 Software de edición

Utilizado para las acciones de diseño y modelado de los elementos que componen cada una de las aplicaciones, así como la generación de documentación:

- Adobe Photoshop CS6
- Microsoft Office 2013

5.1.2.3 Software de consulta

Utilizado a lo largo del proyecto para la consulta de fuentes de información:

- Adobe Reader XI
- Google Chrome

5.2 Plataformas

La creación de un sistema de grandes dimensiones obliga a la utilización de diversos lenguajes de programación, tecnologías y mecanismos para implementar las distintas funcionalidades que se hacen necesarias. A continuación se detallan las diversas plataformas que se van a emplear en el desarrollo del proyecto:

5.2.1 Java

Java es un lenguaje de programación orientado a objetos. Hereda la mayor parte de su sintaxis de los lenguajes de programación C y C++, pero se trata de un lenguaje de nivel más alto que estos dos. Originalmente fue desarrollado por Sun Microsystems y publicado en 1995, pero actualmente pertenece a Oracle Corporation, desde la adquisición de aquella empresa por ésta última.

Java es un lenguaje especialmente eficiente en la creación de aplicaciones cliente-servidor, gracias a la facilidad que presenta para la comunicación entre dispositivos. Además, su integración en páginas web (applets) y su utilización en dispositivos móviles han conllevado un crecimiento exponencial para pasar a convertirse en uno de los lenguajes de programación más utilizados actualmente.

Java se ofrece en dos paquetes posibles:

- **Java Runtime Environment (JRE)**: Permite la ejecución de aplicaciones realizadas en Java.
- **Software Development Kit (SDK)**: Se utiliza para el desarrollo de aplicaciones en Java. La diferencia con el JRE radica en que dispone de utilidades y un compilador.

5.2.2 MySQL

MySQL es el sistema de gestión de bases de datos relacional open source más utilizado del mundo. Al igual que ocurre con Android, se ofrece de manera libre y gratuita, pero para hacer un uso privativo en las aplicaciones se debe adquirir una licencia especial que permita dicha utilización.

El lenguaje MySQL se encarga exclusivamente de la administración de la base de datos, permitiendo realizar consultas, inserciones, modificaciones y demás a través de los comandos necesarios para realizarlas. Al no disponer de interfaz gráfica que facilite su utilización, generalmente se integra con el entorno MySQL Workbench, que permite manejar las bases de datos de forma visual, listar los contenidos y modificar los campos sin necesidad de introducir las sentencias necesarias para ello.

Entre sus características más destacables se puede mencionar su soporte multiplataforma, la posibilidad de uso de triggers y cursores, soporte para SSL (Secure Sockets Layer) y múltiples opciones de back-up.

5.2.3 JSP

JavaServer Pages es una tecnología orientada a la creación de páginas web dinámicas basada en HTML y que utiliza Java para definir la funcionalidad. Al igual que ocurre con Java, tanto las páginas como los servlets se ejecutan en una máquina virtual, lo cual facilita su ejecución en cualquier tipo de dispositivo, siendo únicamente necesario disponer de la máquina virtual para su correcta visualización.

Para desplegar la web desarrollada es necesario disponer de un servidor compatible con la ejecución de servlets, siendo Apache Tomcat el más común.

5.3 Codificación

El sistema se compone de dos aplicaciones. Por un lado, el software para móvil es desarrollado para el Sistema Operativo Android v.4.0 Ice Cream Sandwich, mientras que la aplicación web se monta sobre un servidor Apache Tomcat v7.0.

5.3.1 Aplicación web

La aplicación web se desarrolla haciendo uso de JSP para definir la estructura de la página, y se hace uso de servlets para dotar de contenido dinámico a la misma, realizando las consultas sobre la base de datos de forma segura haciendo uso de dichos servlets.

5.3.1.1 Login

Consiste en una página simple cuyo objetivo es el de dar la posibilidad al usuario de hacer login en el sistema o registrarse en el mismo. Posee dos campos de texto a rellenar con el nombre de usuario y la contraseña. Cuando se pulsa sobre Enviar, se comunica con el servlet *LoginCheck*, donde se validan los datos. En caso de no encontrarse registrado se pasa a la página *Registro*.

5.3.1.2 LoginCheck

Se trata de un servlet donde se validan los datos introducidos por el usuario en el formulario de acceso. Dichos datos se reciben por *post* desde la página anterior, tras lo que se realiza una consulta sobre la tabla Usuarios haciendo uso del nombre de usuario introducido. Si se detecta una entrada, se hace uso de la biblioteca auxiliar BCrypt[5] para calcular la función Hash de la contraseña introducida por el usuario y compararla con el Hash almacenado en la base de datos. Si el resultado de la función Hash coincide, se da acceso al usuario al sistema, antes de lo cual se comprueba si se trata de un usuario Administrador. En caso afirmativo, será redirigido de forma automática a la página *AdminMenuPrincipal*. En caso contrario, accederá a la página *MenuPrincipal*. Para validar al usuario dentro del sistema, se hace uso de una sesión, donde se

[5] Explicada en el apartado 5.4.3 Función Hash para almacenar contraseñas.

almacena el nombre del usuario en el campo *username.* En caso de que el usuario sea administrador, se crea un campo adicional en la sesión llamado *admin*.

Por último, en caso de que el nombre de usuario no se encontrara en la base de datos, o el resultado de la función Hash no coincidiera con la entrada de la tabla Usuarios, se invalida la sesión y se devuelve el control a la página *Login*, añadiendo el mensaje "Usuario o contraseña incorrectos".

5.3.1.3 Registro

La página *Registro* posee un aspecto similar a la página *Login*. Se compone de un formulario de mayor tamaño, donde se insta al usuario a introducir sus datos para darlo de alta en el sistema.

En primer lugar se solicitan los datos de acceso: nombre de usuario, contraseña y repetir contraseña para garantizar que no se ha cometido un error al introducirla.

En segundo lugar se solicitan los datos personales del usuario: nombre y apellidos, DNI, sexo, fecha de nacimiento, dirección, teléfono y e-mail.

Cuando el usuario ha rellenado todos los datos y pulsa sobre Enviar, se contacta con el servlet *RegCheck*, donde se validan los datos y se formaliza el registro.

5.3.1.4 RegCheck

Consiste en un servlet que recibe todos los datos introducidos por el usuario en la página *Registro*. El paso de parámetros se realiza por *post*, y una vez recibidos se realiza una serie de comprobaciones sobre dichos datos antes de proceder a su inserción sobre la base de datos, para lo cual hace uso de una serie de métodos integrados en la misma clase:

- Se comprueba que los campos no están vacíos. En caso contrario, se vuelve a la página *Registro* con el mensaje "Se deben rellenar todos los campos".
- Se comprueba si el nombre de usuario ya se encuentra registrado. En caso de que así sea, se vuelve a la página *Registro* con el mensaje "El usuario introducido ya se encuentra registrado".
- Se comprueba si las dos contraseñas introducidas son iguales. En caso contrario, se vuelve a la página *Registro* con el mensaje "Las contraseñas introducidas no son iguales".
- Se comprueba si la contraseña tiene un mínimo de 8 caracteres. En caso contrario, se vuelve a la página *Registro* con el mensaje "La contraseña debe tener un mínimo de 8 caracteres".

- Se comprueba si la contraseña contiene letras, números y símbolos. En caso contrario, se vuelve a la página *Registro* con el mensaje “La contraseña debe contener números, letras y símbolos”.
- Se comprueba si la fecha introducida sigue un formato de fechas haciendo uso de un *match*. En caso contrario, se vuelve a la página *Registro* con el mensaje “El formato de la fecha debe ser dd/mm/aaaa o dd-mm-aaaa”.
- Se comprueba que los datos no exceden el máximo permitido por la base de datos. En caso contrario, se vuelve a la página *Registro* con el mensaje “Los datos introducidos exceden el tamaño máximo”.
- Se comprueba el formato de la dirección de e-mail. En caso de que no siga la forma de una dirección real, se vuelve a la página *Registro* con el mensaje “La dirección de e-mail introducida no es correcta”.

Si se han superado todas las comprobaciones de forma satisfactoria, se continúa con el proceso de registro. Adicionalmente a los campos rellenados por el usuario, éste posee una serie de campos asignados por el sistema. Tanto el número de cuenta como el CCV se generan de forma aleatoria haciendo uso de la funcionalidad SecureRandom de Java. En caso de que el número de cuenta asignado ya se encuentre utilizado por otro usuario, se genera uno nuevo.

Una vez terminado este procedimiento, los datos del usuario se encuentran listos para ingresarlos en el sistema. Se crea una nueva entrada en la tabla Usuarios y se usa el nombre de usuario introducido como identificador de la entrada. A su vez, se hace uso de la librería BCrypt para aplicar la función Hash a la contraseña introducida por el usuario y almacenar el Hash de dicha contraseña. De la misma forma, se crea una entrada en la tabla Fondos, también haciendo uso del nombre de usuario como identificador de la tabla.

Si el procedimiento concluye de forma satisfactoria, se devuelve el control a la página de Login acompañado del mensaje “Usuario registrado”.

5.3.1.5 MenuPrincipal

Para garantizar la seguridad de las consultas, las páginas se definen a partir de servlets, donde se procesan los datos y consultas necesarias, tras lo cual se genera el código JSP de forma dinámica, incluyendo el contenido resultante de las consultas en el código de la página de forma transparente para el usuario. *MenuPrincipal* realiza una consulta sobre la tabla Usuarios a partir del nombre de usuario almacenado en la sesión. De esta forma, se le permite conocer su nombre real y su sexo, para poder generar el mensaje de bienvenida correspondiente.

Una vez introducido el usuario, la página incluye un pequeño resumen del objetivo del sistema, y ofrece acceso a toda la funcionalidad del mismo:

- Últimos movimientos
- Nueva transferencia
- Información personal
- Salir

El acceso a toda esta funcionalidad está disponible desde todas las vistas una vez se ha hecho login en el sistema.

5.3.1.6 Movimientos

Movimientos se corresponde con la página *Últimos Movimientos*, y ofrece al usuario una tabla donde se listan todos los movimientos de fondos, tanto entrantes como salientes, realizados haciendo uso de su cuenta personal. Para ello se realiza una consulta sobre la tabla Usuarios a partir del nombre de usuario obtenido de la sesión activa. De dicha consulta se obtiene el número de cuenta del mismo, que se emplea para realizar una segunda consulta sobre la tabla Transferencias, buscando entradas que dispongan de dicho número como cuenta de origen o cuenta de destino. Dependiendo de dicha procedencia, se cataloga la transferencia como ingreso o pago. La información que se ofrece al usuario en dicha tabla es la siguiente:

- Transacción: Ingreso o pago.
- Cuenta: Número de cuenta sobre el que se realiza el pago o que realiza el ingreso.
- Importe: Cantidad de dinero transferida.
- Fecha: Fecha de realización de la transacción.
- Concepto: Campo de texto opcional para aclarar la razón de la transferencia.
- Confirmada: Si o No, en función de si ha sido confirmada mediante la aplicación móvil.

Por último, se realiza una consulta sobre la tabla Fondos a partir del nombre de usuario. El resultado es el saldo total de éste. Dicho dato se incluye al final de la tabla.

5.3.1.7 Transferencia

Se corresponde con la página *Nueva Transferencia*. Durante la carga de la página se realiza una consulta sobre la base de datos para comprobar si el identificador del teléfono se ha rellenado. En caso de que no sea así, se informa al usuario que debe registrarse en la aplicación móvil antes de poder realizar una transferencia. Si ya se ha completado el registro, se carga la página de forma normal, ofreciendo al usuario un formulario donde introducir los siguientes datos para formalizar la transferencia:

- Número de cuenta: Cuatro campos de cuatro valores numéricos cada uno.
- Cantidad a transferir: Cantidad en euros que se quiere mover.
- Concepto de la transferencia: Campo de texto para aclarar la razón de la transferencia.

El botón Confirmar pasa dichos datos al servlet TransCheck.

5.3.1.8 TransCheck

Servlet encargado de validar y formalizar la transferencia de fondos de una cuenta a otra. Al igual que ocurría con *RegCheck*, se verifica la validez de los datos introducidos por el usuario:

- Se comprueba que los campos a rellenar no estén vacíos (salvo el campo concepto, que es opcional). En caso contrario, se vuelve a la página *Transferencia* con el mensaje "Se deben rellenar todos los campos".
- Se comprueba que los dígitos que componen la cuenta solamente sean numéricos. En caso contrario, se vuelve a la página *Transferencia* con el mensaje "La cuenta solamente puede estar compuesta de números".
- Se comprueba el tamaño de la cuenta introducida, que debe ser de cuatro campos de cuatro cifras cada uno. En caso contrario, se vuelve a la página *Transferencia* con el mensaje "La cuenta introducida es incorrecta".
- Se comprueba que la cantidad de dinero introducida sea numérica y no sea negativa. En caso contrario, se vuelve a la página *Transferencia* con el mensaje "La cantidad introducida debe ser numérica".

Si se han superado todos los controles de forma satisfactoria, se genera un identificador único y aleatorio para la transferencia, haciendo uso de SecureRandom. Tras esto, se realiza una consulta sobre la tabla Usuarios y se recogen el número de cuenta y el identificador único del teléfono, tras lo que se obtiene la fecha actual haciendo uso del método correspondiente. Una vez se tienen todos los datos, se realiza una inserción sobre la tabla Transferencias, donde se introducen todos los datos anteriores, acompañados de un "n" en el campo Confirmada, para indicar que aún no se ha confirmado la transferencia.

Una vez finalizada la inserción, se redirige a la página *TransConfirma*.

5.3.1.9 TransConfirma

Se trata de la página donde radica la seguridad del sistema, ya que se hace uso del segundo mecanismo de verificación de la identidad del usuario. De la misma forma, y por la misma razón, se trata de la página donde se debe llevar a cabo la interacción con el dispositivo móvil.

Al cargar, simplemente muestra un código QR acompañado de un texto que solicita al usuario que lo escanee con su aplicación móvil. Dicho código QR se carga como una imagen en la página, pero en su generación interactúan el servlet *GeneraQR* y la librería *CriptoRSA*, el primero para generar el código, y el segundo para cifrar el contenido del mismo.

5.3.1.10 GeneraQR

Servlet que permite la generación de un código QR a partir de una cadena de texto String. Este servlet recibe como variables el nombre del usuario, el identificador único de la transferencia y el importe de la misma, y genera una nueva cadena con dichos tres parámetros concatenados.

Una vez hecho esto, se realiza una consulta sobre la tabla Claves utilizando el nombre de usuario como clave, lo cual devuelve la clave pública del usuario. Teniendo la clave, se hace uso de la función Cifra de la biblioteca *CriptoRSA*, la que, al pasarle la clave pública y la cadena de texto a cifrar, devuelve el texto cifrado con la clave en formato String. La cadena resultante se utiliza para generar el código QR en un formato de imagen PNG de 300 x 300px, lo cual se inserta en el stream de salida para que pueda ser interpretado por la página anterior como una imagen simple.

5.3.1.11 CriptoRSA

Se trata de una biblioteca que permite el cifrado y descifrado de mensajes haciendo uso del algoritmo RSA. Dispone de cuatro métodos:

- **Claves**: Genera un par de claves RSA de tipo KeyPair. Dicho par de claves contiene una clave pública y una clave privada.
- **Cifra**: Recibe como parámetros una cadena de entrada a cifrar y una clave de cifrado. La cadena de entrada se convierte a un array de bytes UTF-8, tras lo cual se inicializa el método en modo cifrado haciendo uso de RSA. El array de bytes correspondiente se convierte a String haciendo uso de la herramienta Arrays.toString() para su manejo más sencillo (codificación en un QR, almacenamiento en el dispositivo y demás).
- **Descifra**: Recibe como parámetros de entrada una cadena de texto cifrado y una clave para descifrar la cadena. La cadena de entrada contiene los bytes convertidos a String, de manera que se hace uso del método convierteBytes para obtener el array de bytes original. Tras esto, se inicializa el método en modo descifrado haciendo uso de RSA.

- **ConvierteBytes**[6]: La función Arrays.toString() convierte el array de bytes en texto, utilizando el propio array como entrada en lugar del contenido del mismo, de manera que se hace necesario el uso de un método adicional para su conversión a bytes de nuevo. Para ello, se dispone de un bucle que recorre la cadena de texto y utiliza la coma como *token* de separación, volviendo a generar cada uno de los caracteres originales.

Esta biblioteca también se encuentra incluida en la aplicación móvil, como se verá más adelante.

5.3.1.12 DatosPersonales

Conforma la página *Información Personal*. Cuando se carga dicha página, se realiza una consulta sobre la tabla Usuarios a partir del nombre de usuario obtenido de la sesión. De dicha consulta se obtienen todos sus datos personales, además del número de cuenta del usuario y el código CCV. Estos dos últimos valores se muestran en la parte superior de la página a nivel informativo, de manera que no pueden ser modificados por el usuario. Debajo se encuentra una tabla que contiene un formulario similar al de la página de *Registro*, donde cada uno de los campos se encuentra relleno con los datos proporcionados por el usuario. Dichos datos pueden ser modificados, y para que los datos alterados tengan efecto permanente se debe presionar el botón Modificar, que hace uso del servlet *DatosCheck* para formalizar dichas modificaciones.

5.3.1.13 DatosCheck

Se trata de un servlet cuyo funcionamiento es similar a *RegCheck*. Se reciben por *post* los parámetros del formulario y se realizan las mismas comprobaciones sobre los datos que en el caso anterior. En el momento de realizar la inserción sobre la base de datos, se realiza un *update* utilizando como clave el nombre de usuario. Si el procedimiento se completa sin problemas, se redirige a la vista de *Menú Principal* con el mensaje “Cambios Realizados”.

5.3.1.14 Logout

Cuando el usuario pulsa sobre el botón Salir, se hace uso del servlet *Logout* para invalidar la sesión del mismo. Una vez invalidada, se devuelve el control a la página Login para permitir un nuevo acceso a la página.

5.3.1.15 AdminMenuPrincipal

El funcionamiento es similar a la página *Menú Principal* para un usuario normal del sistema, con la diferencia de que en la sesión se comprueba que también disponga del campo *admin*. De la

[6] Este método se hace necesario dado que las funciones de conversión de bytes a String de Java pueden dar problemas con algunos caracteres, que a veces toman su valor ASCII positivo y a veces su valor negativo, lo cual derivaba en errores a la hora de descifrar el mensaje original.

misma forma, la funcionalidad a la que tiene acceso es distinta, orientada a las acciones que puede realizar el administrador sobre el sistema:

- Lista de usuarios
- Lista de transferencias
- Nuevo usuario
- Salir

Al igual que en el caso anterior, desde cualquier página consultada por el administrador, se tiene acceso a toda esta funcionalidad.

5.3.1.16 AdminUsuarios

Se corresponde con la página *Lista de usuarios*, y ofrece un listado completo de todos los usuarios del sistema, para lo cual se realiza una consulta completa sobre todos los registros de la tabla Usuarios. Dichos usuarios se encuentran catalogados en una tabla e identificados por su nombre de usuario. Adicionalmente se ofrecen el nombre y los apellidos del mismo, su dirección de email y su número de cuenta. Desde dicha vista se ofrece acceso a la consulta de los datos completos del usuario haciendo click sobre el nombre de usuario, lo cual referencia a la página *AdminDatosUsuario*.

5.3.1.17 AdminDatosUsuario

Se trata de una página que recibe como parámetro el nombre del usuario sobre el que se quiere realizar la consulta de los datos, de manera que realiza una consulta sobre la tabla Usuarios a partir de su nombre de usuario. La apariencia es la misma que la de la página *DatosPersonales* para el usuario normal, y ofrece todos los datos personales del usuario en un formulario donde se permite su modificación. Al pulsar sobre el botón Modificar, se comunica con la página *AdminDatosCheck*, donde se formaliza la modificación de dichos datos.

5.3.1.18 AdminDatosCheck

El funcionamiento es el mismo que el del servlet *DatosCheck*. Se reciben los datos del formulario por *post* y se realizan las comprobaciones sobre dichos datos para garantizar que son correctos y no presentan ningún riesgo para la integridad del sistema. Una vez comprobado, se insertan en la base de datos y se redirecciona a la página *AdminMenuPrincipal* con el mensaje “Cambios realizados”.

5.3.1.19 AdminTransferencias

Se corresponde con la página *Lista de transferencias*. Durante la carga de la página se realiza una consulta completa sobre todos los registros de la tabla Transferencias y se rellena una tabla

donde se ofrecen el número de cuenta de origen de la transferencia, el número de cuenta de destino, el importe de la misma, el concepto y si se encuentra confirmada o no.

5.3.1.20 AdminNuevoUsuario

La página *Nuevo Usuario* permite dar de alta un nuevo usuario desde cero. El formulario a rellenar es similar al que se ofrece al usuario en la página de *Registro*, y cuando se pulsa sobre el botón Aplicar, se redirige la acción al servlet *AdminNuevoUsuarioCheck*.

5.3.1.21 AdminNuevoUsuarioCheck

El funcionamiento es similar al del servlet *RegCheck* para el registro de un usuario normal. Se reciben los parámetros por *post*, se realizan las comprobaciones sobre los datos y, si no se detectan problemas sobre dichos datos, se realizan las inserciones sobre las tablas Usuarios y Fondos. Una vez formalizado el registro, se redirige al Menú Principal con el mensaje "Usuario registrado".

5.3.2 Servlets de interacción

Se trata de una serie de servlets localizados en el servidor donde se aloja la aplicación. Se utilizan para permitir la interacción entre la aplicación web y la aplicación móvil, y la utilización de la base de datos por ésta última sin poner en peligro la integridad de ésta al no realizarse directamente la conexión desde el propio dispositivo.

5.3.2.1 AndroidConn

Primera conexión del dispositivo con el servidor. Se recibe el identificador único del dispositivo por *post*, tras lo que se realiza una consulta sobre la tabla Usuarios de la base de datos para comprobar si dicho parámetro se encuentra registrado. En caso afirmativo, se devuelve al dispositivo la cadena "registrado" como respuesta. En caso contrario, se devuelve la cadena "no_registrado".

5.3.2.2 AndroidReg

Servlet que permite al usuario completar el registro en la aplicación. Dicho registro se completa cuando el usuario se instala la aplicación en su terminal móvil y hace login con sus datos de acceso. Se reciben por *post* el identificador único del dispositivo, el nombre de usuario, la contraseña y la clave pública del usuario.

En primer lugar, se realiza una consulta sobre la tabla Usuarios con el nombre de usuario. En caso de encontrarse una coincidencia, se calcula el Hash de la contraseña introducida haciendo uso de la librería BCrypt, y se compara con el Hash almacenado en la base de datos. Si coincide con éste, el usuario ha verificado su identidad, con lo que se procede a formalizar el registro.

Para ello se realiza un *update* sobre la tabla Usuarios de la base de datos para introducir el identificador único del teléfono, y un *insert* en la tabla Claves donde se relacionan el nombre de usuario con su clave pública para permitir el cifrado de mensajes.

Si el procedimiento se completa sin problemas, se devuelve un *ok* como respuesta a la aplicación móvil.

5.3.2.3 AndroidConf

Servlet que recibe la respuesta de la aplicación móvil a la confirmación de la transferencia codificada en un QR. Dicha respuesta se encuentra cifrada con la clave pública del servidor, de manera que sea éste el único que puede leer su contenido.

La respuesta de la aplicación móvil se recibe por *post*, como una cadena "cifrado" de tipo String. Para descifrar el contenido del mensaje, se consulta el documento local Claves, donde se encuentran cifradas por AES tanto la clave pública como la privada del servidor. Se realiza una consulta para obtener la clave privada, lo cual devuelve el módulo y el exponente de la misma en formato BigInteger. Haciendo uso de KeyFactory e instanciado con RSA, se genera la clave privada, tras lo que se hace uso de la función Descifra de la librería CriptoRSA para obtener el texto en claro.

A partir del texto en claro, se obtienen el identificador del teléfono, el identificador de la transferencia, la cantidad de dinero y el nombre de usuario enviados desde la aplicación móvil. En primer lugar, se comprueba que el identificador del dispositivo sea el correspondiente al usuario. Una vez hecho esto, se verifica que el resto de datos sean los correspondientes a la transferencia solicitada.

Confirmados todos los parámetros de la transferencia, se procede a insertar los registros en la base de datos. En primer lugar se realiza un *update* en la tabla Transferencias para reflejar la transferencia como confirmada. En segundo lugar se actualiza la tabla Fondos para realizar el movimiento de dinero de una cuenta a otra.

Una vez se ha completado todo el proceso, se devuelve un *ok* como respuesta a la aplicación Android.

5.3.3 Aplicación móvil

La aplicación móvil se desarrolla para el sistema operativo Android 4.0 Ice Cream Sandwich. El diseño de la misma es elástico y se adapta a la pantalla de cualquier dispositivo, y está preparado para visualizarse de forma correcta sobre Smartphones, dispositivos de alta resolución y tablets.

5.3.3.1 MainActivity

Actividad que arranca con el inicio de la aplicación. Muestra una pantalla de carga al usuario mientras realiza las operaciones necesarias para poner en funcionamiento la aplicación. Dispone de dos métodos:

- **Identificador**: Genera un identificador único para el dispositivo a partir de los siguientes parámetros:
  - IMEI
  - Identificador del terminal
  - Identificador de Android
  - Dirección MAC

  Una vez calculados todos, se concatenan en una cadena y se calcula el MD5 de la misma. El resultado es el identificador único del dispositivo con el que se registra dentro del sistema.
- **Consulta**: Realiza una llamada a una dirección pasada por parámetro y le pasa por post un segundo parámetro. La respuesta recibida de la página se procesa como String y se devuelve como resultado de la función.

La actividad crea un nuevo hilo donde lanza la consulta al servlet *AndroidConn*, y le pasa por parámetro en identificador único de terminal. Si la respuesta del servlet es "registrado", el proceso de registro ya se ha completado anteriormente, de manera que se pasa directamente al *Menú principal* de la aplicación. Si por el contrario se recibe "no_registrado", se debe registrar al usuario, de manera que se llama a la actividad *RegistroActivity*.

5.3.3.2 RegistroActivity

Esta actividad es accedida cuando el usuario aún no ha completado el proceso de registro del sistema. Se solicita al usuario introducir el nombre de usuario y la contraseña con que se ha registrado en la aplicación. Una vez se rellenan dichos campos, se realiza una llamada a un servlet similar a la actividad anterior.

En primer lugar, como el usuario aún no dispone claves para cifrar los mensajes, se hace uso de la función Claves de la librería CriptoRSA, lo cual devuelve el par de claves de tipo KeyPair a partir del cual se obtienen la clave pública y la clave privada.

Tras esto, se contacta con el servlet AndroidReg para completar el registro, al cual se pasan el identificador del dispositivo, el usuario y la contraseña introducidos y la clave pública recién generada como un String que contiene el array de bytes de la misma.

Si el proceso de registro se ha completado satisfactoriamente, se almacena al clave privada del usuario en el dispositivo. Para almacenar la clave de manera segura, se cifra la misma haciendo uso de un algoritmo AES y utilizando como contraseña la clave introducida por el usuario. Para ello se hace uso de la clase CriptoAES, similar a CriptoRSA pero orientado al uso de algoritmo AES. Para su correcto funcionamiento dispone de un método adicional generaClave, que recibe una clave en String y devuelve una clave adaptada para AES en formato SecretKeySpec. La cadena cifrada resultante se almacena en las propiedades de la aplicación, de manera que no pueda ser accedida de manera sencilla por el usuario ni por otras aplicaciones.

Una vez se completan todos los procedimientos anteriores, se accede al menú principal de la aplicación.

5.3.3.3 MenuPrincipalActivity

La actividad más sencilla de la aplicación. Una vez validado el usuario, se accede a esta aplicación donde se ofrece la posibilidad de realizar una captura para validar un código QR. Presenta el logo del sistema en la parte superior, y un botón de gran tamaño en el centro para acceder a dicha funcionalidad.

Cuando se presiona dicho botón, se accede a la actividad de captura.

5.3.3.4 CapturaActivity

Actividad que hace uso de la cámara del dispositivo a través de la clase CameraPreview y la librería ZBar para buscar un código QR en las imágenes capturadas por la cámara. Cuando se detecta un código QR, se escanea su contenido de manera automática y se accede a la siguiente actividad, *ResultadoActivity*, donde se procesa su contenido.

5.3.3.5 ResultadoActivity

Se trata de la actividad donde se procesa el contenido del código QR escaneado y se comunica con el servlet correspondiente.

Al iniciar la actividad, se genera un Alert para solicitar al usuario que introduzca la constraseña de su cuenta. Una vez obtenido, se carga la clave privada cifrada de las propiedades de la aplicación y se descifra haciendo uso del método Descifra de la clase CriptoAES, al que se pasa por parámetro tanto la cadena cifrada como la contraseña. A partir de la cadena descifrada, se genera la clave de tipo PrivateKey, y se utiliza para descifrar el contenido del mensaje codificado en el código QR. Para ello, se hace uso del método Descifra de la clase CriptoRSA, al que se pasan el código obtenido del QR y la clave privada del usuario. El resultado es una cadena de texto que

contiene el identificador de la transferencia, la cantidad de dinero transferido y el nombre del usuario que solicita la transferencia.

Tras esto, se crea un nuevo hilo donde se solicita la llamada al servlet *AndroidConf*. Para ello, se carga la clave pública del servidor a partir del módulo y el exponente, y se genera una nueva cadena de texto donde, además de los parámetros obtenidos del código QR, se añade el identificador único del teléfono. Dicha cadena se cifra con la clave pública del servidor haciendo uso del método Cifra de la clase CiptoRSA y se envía por *post* al servlet anteriormente mencionado.

Si tras realizar las verificaciones, el servlet responde con un *ok*, se llama a la actividad *ConfirmaActivity* pasándole un mensaje "confirmada" en el *intent* de la llamada. Si por el contrario no se superan las verificaciones, se llama a dicha actividad pero se pasa un mensaje "rechazada" en el *intent*.

5.3.3.6 ConfirmaActivity

Al igual que *MenuPrincipal*, se trata de una actividad muy sencilla, orientada a informar al usuario del resultado del proceso. En caso de que se reciba un mensaje "confirmada" en la llamada a la actividad, se muestra el mensaje "Transferencia confirmada" acompañado de un tick verde. Si por el contrario se recibe el mensaje "rechazada", se muestra el mensaje "Transferencia rechazada", acompañado de un aspa roja.

5.4 Mecanismos de seguridad

A continuación se explican los diversos mecanismos que se han empleado para garantizar la seguridad del sistema.

5.4.1 Protocolo SSL/TLS

Para evitar que un atacante pueda interceptar información mediante la escucha de la comunicación entre el cliente y el servidor, se establece un protocolo de comunicación por SSL, obteniendo una conexión segura donde se cifra el contenido antes de enviarse y se descifra a su llegada al destino.

Para ello, en primer lugar, se generan tanto las claves y los certificados, como el propio almacén de claves haciendo uso de la herramienta KeyTool de Java. Tras esto, en el fichero server.xml del servidor, se habilita la comunicación a través de SSL:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
      maxThreads="150" scheme="https" secure="true"
      clientAuth="false" sslProtocol="TLS"
      keystoreFile="          "
      keystorePass="          "/>
```

Por último, para obligar a que todas las comunicaciones se realicen siempre por SSL, se añade la siguiente sentencia en el fichero *web.xml* de configuración de la página web:

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>securedapp</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

De esta forma, todas las comunicaciones que se realicen con el servidor se establecerán siempre a través de un canal seguro.

5.4.2 Sesiones

Las sesiones se utilizan para verificar la identidad del usuario a lo largo de todo el tiempo de uso del sistema, desde que hace login hasta que hace logout. El usuario queda identificado mediante su nombre de usuario, de manera que, de ser necesario, se pueden realizar consultas sobre la base de datos para conocer más datos de dicho usuario.

El procedimiento para hacer uso de sesiones consiste en su habilitación en el servlet encargado de procesar el login:

```
session.setAttribute("username", username);
```

Su captura y verificación en cada una de las páginas a las que puede acceder el usuario:

```
HttpSession session = request.getSession(true);
String username = session.getAttribute("username").toString();
```

Y su invalidación, una vez ha finalizado su uso, en la pantalla de logout:

```
HttpSession session = request.getSession(true);
session.removeAttribute("username");
session.invalidate();
```

De esta forma, se evita la suplantación de la identidad del usuario por un posible atacante.

5.4.2.1 Timeout de la sesión

Para garantizar que la sesión no se queda abierta si el usuario no cierra la aplicación siguiendo el procedimiento de logout, o se deja la página abierta, se establece un *timeout* o tiempo máximo de sesión, de manera que si el usuario no realiza ninguna acción sobre la página en dicho tiempo, la sesión se cierra de forma automática.

Para ello, se añade la siguiente cadena en el fichero web.xml:

```
<session-config>
      <session-timeout>15</session-timeout>
</session-config>
```

Como se puede observar, se establece un tiempo máximo de sesión de 15 minutos.

5.4.3 Función Hash para almacenar contraseñas

La función Hash se utiliza para almacenar las contraseñas de un sistema ya que, al ser una función de un único sentido, no se puede calcular la cadena original a partir del Hash.

Para implementar la función Hash se hace uso de la librería BCrypt, que es un tipo de cifrado que hace uso de un *salt* para evitar ataques que hagan uso de tablas arcoíris, y el algoritmo de cifrado es suficientemente complejo para que requiera de un tiempo de procesamiento alto. Dicho tiempo no será apreciable para un usuario que haga login sobre la aplicación, pero resultará enormemente complejo para un atacante que quiera realizar un ataque de fuerza bruta, donde se requiere el cifrado constante de contraseñas.

Para implementar BCrypt en la aplicación, se va a hacer uso de la librería JBCrypt, que consiste en la aplicación del algoritmo utilizado por BCrypt en Java. Para ello se importa la clase BCrypt.java en el proyecto y, a través de la siguiente cadena, se genera la contraseña cifrada:

```
String shaPass = BCrypt.hashpw(password, BCrypt.gensalt());
```

Después basta con insertar la cadena resultante en la base de datos como una cadena de texto normal.

Para hacer login se utiliza la función *checkpw()*, implementanda también en la librería BCrypt, para comprobar la coincidencia de ambas contraseñas.

5.4.4 Cifrado de mensajes con RSA

RSA es un algoritmo de cifrado que hace uso de una clave pública y una clave privada para el cifrado de mensajes. La seguridad de este algoritmo radica en el problema de la factorización de números enteros. Los mensajes enviados se representan mediante números, y el funcionamiento se basa en el producto, conocido, de dos números primos grandes elegidos al azar y mantenidos en secreto.

Para cifrar un mensaje, el emisor utiliza la clave pública del receptor, de manera que éste pueda utilizar su clave privada para descifrar el contenido del mismo. Si por el contrario quiere firmar el mensaje, utilizará su propia clave privada para cifrar el mensaje. De esta forma, el receptor podrá descifrar el contenido del mismo haciendo uso de la clave pública del emisor.

Para facilitar el uso del algoritmo RSA, se desarrolla la librería CriptoRSA, que permite tanto la generación del par de claves, como el cifrado y el descifrado de mensajes.

La implementación que se sigue en el proyecto cifra los mensajes intercambiados entre la aplicación web y la aplicación móvil. En un primer momento, el contenido del código QR (identificador de la transferencia, cantidad a transferir y nombre de usuario) es cifrado por el servidor haciendo uso de la clave pública del usuario. Una vez cifrado, se genera el código QR con la cadena resultante. Cuando el usuario escanea el código con su aplicación móvil, se utiliza la clave privada del usuario para descifrar el contenido del mismo. Se obtienen los parámetros mencionados, se añade el identificador del teléfono y la cadena resultante se cifra de nuevo con la clave pública del servidor para ser enviada a través de la red al servlet correspondiente. Cuando dicha cadena es recibida por el servidor, se hace uso de la clave privada del mismo para descifrar su contenido, se procesan los datos enviados por la aplicación móvil y se envía la respuesta en función de los datos recibidos.

5.4.5 Cifrado de información con AES

AES es un esquema de cifrado por bloques que permite claves de 128, 192 y 256 bits. Actualmente se trata de uno de los algoritmos más populares en criptografía simétrica.

Para su utilización, se desarrolla la librería CriptoAES, similar a CriptoRSA pero orientada al cifrado de información haciendo uso de AES.

Su uso en el sistema se centra en el cifrado de la clave privada del cliente para su almacenamiento seguro en el dispositivo. La clave se cifra utilizando como contraseña la clave de la cuenta del usuario, de manera que cuando se va a hacer uso de la clave privada (para descifrar el código QR), se solicita al usuario que introduzca dicho código. Haciendo uso de la librería anterior, se descifra la clave y se permite su utilización.

De la misma forma, el almacenamiento de la clave privada del servidor también se cifra haciendo uso de AES, y se almacena sobre un documento almacenado de forma local en el propio servidor.

5.4.6 Inyección de código

Uno de los ataques más comunes, dada su relativa facilidad, es la inyección de código en los formularios habilitados para el usuario. Se trata de un ataque que, en caso de no ser prevenido, puede suponer riesgo grave que pone en peligro la integridad de los datos de todo el sistema.

Dado que las consultas sobre la base de datos se realizan desde Java, para evitar los efectos de este ataque, se hace uso de la interfaz *PreparedStatement*, que permite realizar la inserción de los datos introducidos por el usuario de forma separada al resto de la sentencia:

```
PreparedStatement st = conn.prepareStatement
      ("SELECT * FROM usuarios WHERE username=?");
st.setString(1, username);
ResultSet res = st.executeQuery();
```

De esta forma, al no insertarse la cadena introducida por el usuario en la propia consulta SQL, no se ejecuta por el intérprete, sino que se separa el contenido introducido y se procesa de forma independiente. Si se introduce una cadena SQL ejecutable en dicha sentencia, será interpretada como texto plano sin causar problema alguno al sistema.

5.4.7 Secuencias de comandos en sitios cruzados

Se trata de otro ataque común en páginas web. Conocido comúnmente como XSS, se trata de una vulnerabilidad que permite a un atacante inyectar un código autoejecutable (generalmente JavaScript) en una web, de manera que sea ejecutado por los exploradores de todos los visitantes de dicho dominio.

Para evitar este ataque, se hace uso de JTSL, y mediante *<c:out>* se filtra el contenido introducido por el usuario antes de ser mostrado por pantalla.

```
Bienvenido, <c:out value="${param.nombre}" />
```

De esta forma, si el atacante incluye código en campos de texto que puedan ser procesados por el explorador, dicho código no será ejecutado, sino tratado como texto plano.

5.4.8 Acceso seguro a base de datos

El acceso a bases de datos online desde un dispositivo móvil puede suponer un grave riesgo para la integridad de dicha base de datos. Las técnicas de decompilación de paquetes y aplicaciones pueden poner en peligro las credenciales de acceso integradas dentro de la propia aplicación, incluso empleando mecanismos de ofuscación del código.

Para evitar la integración de la conexión con la base de datos en la propia aplicación, se hace uso de una serie de servlets alojados en el servidor cuyo objetivo es realizar las conexiones con la base de datos solicitadas por la aplicación y devolverle las respuestas generadas. De esta forma, la aplicación móvil únicamente puede solicitar la realización de las acciones sobre las tablas correspondientes a un agente externo, de manera que no tiene ningún control sobre la base de datos.

Los servlets correspondientes se listan en el apartado 5.3.2 Servlets de interacción.

5.4.9 Identificador único del terminal

Para garantizar la seguridad de las transferencias, estas se confirman haciendo uso de la aplicación móvil, y para que cada usuario tenga asociado un único dispositivo móvil se hace uso de un identificador que registra cada dispositivo de manera única, de manera que el identificador de dos usuarios distintos no puede ser nunca el mismo.

Para generar dicho identificador se desarrolla la función Identificador, que hace uso de los siguientes parámetros para generar una cadena única:

- **IMEI**: Código USSD único para cada dispositivo, pre-grabado en los teléfonos móviles GSM.
- **Identificador del terminal**: Código calculado a partir de la concatenación de información técnica del dispositivo.
- **Identificador de Android**: Código identificador único asociado a cada uno de los dispositivos Android.
- **Dirección MAC**: Código identificador de 48 bits que corresponde de forma única a una tarjeta o dispositivo de red.

A la cadena resultante de la concatenación de dichos parámetros se le aplica el algoritmo de codificación MD5 de 128 bits para mayor seguridad y un manejo más sencillo. El resultado es una cadena conformada por 32 caracteres hexadecimales que identifica de manera única al dispositivo.

5.5 Interfaz

La interfaz del sistema recoge las distintas vistas, la comunicación entre estas y los elementos de interacción que presenta la aplicación para garantizar una interacción del usuario con el sistema correcta y eficiente.

5.5.1 Interfaz web

El sistema se subdivide en un apartado web y una aplicación móvil. La interfaz del sistema web se conforma a través de unos ficheros base con estructura JSP que dan forma a la página sobre los cuales se construyen las distintas vistas de la aplicación, con la funcionalidad correspondiente.

5.5.1.1 Login

Primera vista que se observa al acceder al sistema. Permite a un usuario ingresar en el mismo con sus datos de acceso. Dichos datos son los que ingresa en el formulario de registro la primera vez que accede al sistema.

Desde esta vista se ofrece acceso al formulario de Registro (sin hacer login) y al Menú principal (haciendo login).

5.5.1.2 Registro

Pantalla de Registro que permite a un usuario darse de alta en el sistema. Se solicitan los datos de acceso de la sesión del usuario y los datos personales del mismo para ingresarlos en la base de datos.

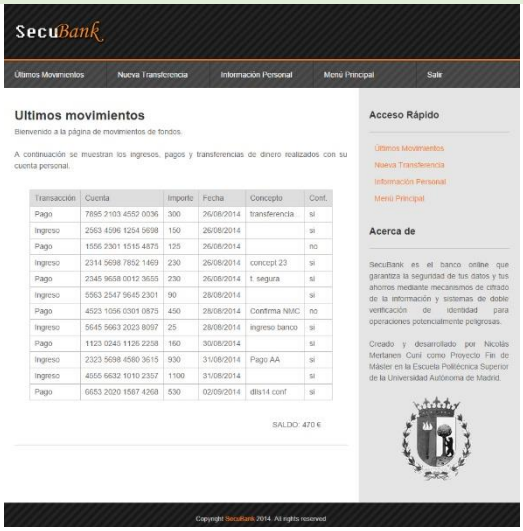
5.5.1.3 Menú principal

El menú principal de la aplicación presenta la bienvenida al usuario al sistema y ofrece acceso a toda la funcionalidad del mismo.

La bienvenida es acompañada por un mensaje donde se recoge el objetivo perseguido por el proyecto.

Se puede apreciar la presencia de un menú superior de acceso a toda la funcionalidad y un menú lateral de acceso rápido.

5.5.1.4 Últimos movimientos



Página que ofrece un listado de los movimientos de fondos entrantes y salientes realizados desde la cuenta del usuario.

Se muestran los detalles de cada una de las transacciones y la confirmación de si dicho movimiento se encuentra confirmado o no.

En la parte inferior de la tabla se ofrece el saldo total de la cuenta del usuario.

5.5.1.5 Nueva transferencia

Permite a un usuario realizar una transferencia de dinero de su cuenta personal a la cuenta de otro usuario. Para realizar dicha transferencia debe introducir el número de cuenta sobre el que transferir el dinero, la cantidad a transferir y, opcionalmente, el concepto de la transferencia.

Una vez rellenados los campos, se redirige a la página que permite confirmar la transferencia a través del código QR correspondiente.

5.5.1.6 Confirmar transferencia

Ofrece al usuario el código QR que permite confirmar la transferencia. Dicho código QR contiene la información necesaria para realizar la confirmación cifrada con la clave pública del usuario, de manera que solamente puede ser desbloqueado desde el dispositivo móvil de éste.

5.5.1.7 Información personal

Presenta un formulario similar al formulario de registro de la aplicación. Permite al usuario consultar los datos con los que se encuentra registrado en el sistema y le da la posibilidad de modificar dichos datos en caso de ser erróneos.

Adicionalmente, muestra el número de cuenta del usuario y el código CCV, generados por el sistema durante el proceso de registro.

5.5.1.8 Menú principal (Administrador)

Menú principal para el administrador del sistema. En apariencia similar al anterior, pero ofrece acceso a nueva funcionalidad, restringida a un usuario normal:

- Listado completo de usuarios
- Listado completo de transferencias
- Dar de alta a un nuevo usuario

5.5.1.9 Lista de usuarios (Administrador)

Ofrece un listado completo de usuarios registrados en el sistema, acompañado de un resumen de sus datos personales más relevantes.

Presionar sobre el nombre del usuario da acceso a la relación completa de datos personales del usuario correspondiente, permitiendo su modificación.

5.5.1.10 Datos personales del usuario (Administrador)

Ofrece al Administrador del sistema los datos personales con los que se ha registrado el usuario seleccionado en el sistema.

Ofrece la posibilidad de modificar dichos datos en caso de detectarse anormalidades o errores.

Al igual que en la vista del usuario, muestra el número de cuenta y el código CCV de dicho usuario.

5.5.1.11 Lista de transferencias (Administrador)

Presenta una tabla donde se recoge el listado completo de transferencias realizadas sobre el sistema.

Cada una de las entradas recoge la cuenta de la cual se substrae el dinero, la cuenta que lo recibe, el importe transferido, la fecha de la transferencia, el concepto de la misma y si dicha transferencia se encuentra confirmada.

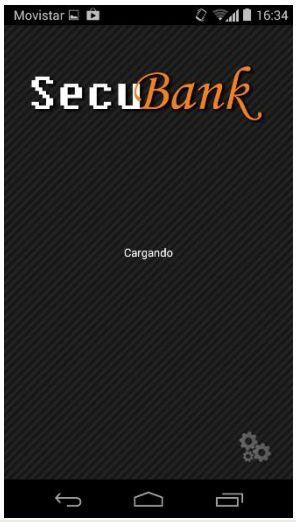
5.5.1.12 Nuevo usuario (Administrador)

Permite al Administrador dar de alta un nuevo usuario en el sistema.

Presenta un formulario similar al de la pantalla de registro de nuevos usuarios.

5.5.2 Interfaz móvil

La interfaz de la aplicación móvil está conformada por una serie de layouts, encargados de dar forma al sistema. Dichos layouts están definidos en una serie de ficheros XML, que son los encargados de determinar la localización de cada uno de los componentes de las vistas.



5.5.2.1 Main

Pantalla de carga de la aplicación. Se comprueba si el usuario ha finalizado el registro en el sistema. En caso de haberse completado, se redirige al Menú principal de la aplicación. En caso contrario, se accede a la pantalla de Registro de la misma.

5.5.2.2 Registro

Pantalla de registro de la aplicación. Se solicita al usuario que introduzca su nombre de usuario y su contraseña para finalizar el registro en el sistema. Debe haberse registrado con anterioridad en la aplicación web para poder finalizar el proceso de registro.

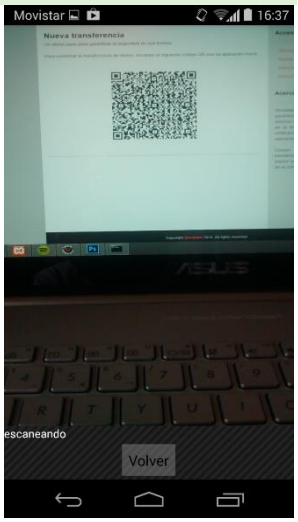
5.5.2.3 Menú principal

Pantalla a la que accede el usuario una vez se ha completado el registro en la aplicación. Permite acceder a la funcionalidad de captura de códigos QR y a la posterior verificación de las transacciones realizadas.

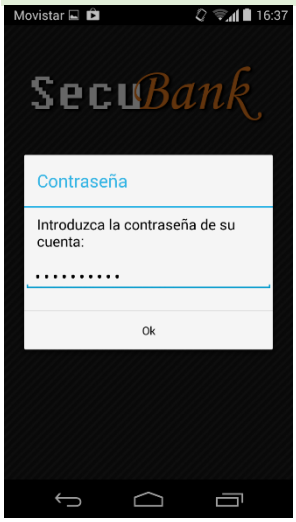
5.5.2.4 Captura

Pantalla que permite al usuario escanear un código QR haciendo uso de la cámara de su dispositivo móvil. El programa detecta de manera automática la presencia de un código QR y procesa el contenido del mismo, de manera que el usuario no tiene que realizar acciones adicionales sobre esta pantalla.

El botón Volver regresa a la pantalla de Menú principal.



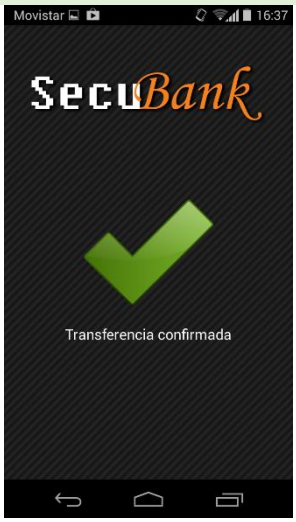
5.5.2.5 Resultado



Pantalla que solicita al usuario su contraseña para verificar el contenido del código QR. La contraseña es necesaria dado que la clave privada del usuario se encuentra cifrada mediante un algoritmo AES donde dicha cadena representa la clave de cifrado.

Una vez introducido, se utiliza la clave privada para descifrar el contenido y procesar la respuesta.

5.5.2.6 Confirmación



Pantalla que presenta al usuario el resultado de la operación de confirmación. En caso de haberse confirmado la transferencia por parte del servidor, se presenta un mensaje de "Transferencia confirmada". En caso contrario, se muestra "Transferencia rechazada".

5.5.3 Diagrama de navegación

Figura 5-1: Diagrama de navegación

Conclusiones

El objetivo principal perseguido con la realización de un proyecto de finalización de máster es la puesta en práctica de los conocimientos adquiridos a lo largo del mismo. De esta forma, el proyecto desarrollado persigue la implementación de un sistema centrado de manera estricta en la seguridad, haciendo uso de diversos mecanismos para garantizar tanto la identidad del usuario como la privacidad de sus datos. De la misma manera, se hace uso de los conocimientos de redes y dispositivos móviles para permitir la interacción entre las dos aplicaciones, y se garantiza la usabilidad del sistema de cara a su utilización por el usuario, siendo este uno de los principales objetivos. Al mismo tiempo se realiza un seguimiento del proyecto exhaustivo y detallado, recogiendo toda la documentación desarrollada en el proceso de estudio e implementación del mismo.

El sistema desarrollado permite a un usuario darse de alta en la aplicación y, además de consultar toda la información relacionada con su cuenta, realizar trasferencias de forma segura haciendo uso de un sistema de autenticación de dos factores mediante el uso del Smartphone.

Como se ha podido observar, se han cumplido cada uno de los requisitos propuestos en un principio, pudiendo concluir que el sistema se ha finalizado de manera plenamente satisfactoria.

Por otro lado, como conclusiones de carácter personal, reconocer que la finalización del actual proyecto ha supuesto un gran esfuerzo, pues todos los problemas encontrados no han hecho sino obligar a realizar una investigación más profunda y elaborada para dar con la solución, conllevando una mayor satisfacción y un incremento de los conocimientos adquiridos.

Nicolás Mertanen Cuní

Colmenarejo, a 7 de septiembre de 2014

[1] M. A. Laborie Iglesias, «La Evolución del Concepto de Seguridad» 2011. [En línea]. URL: http://www.ieee.es/Galerias/fichero/docs_marco/2011/DIEEEM05-2011EvolucionConceptoSeguridad.pdf. [Último acceso: 06 2014].

[2] P. Celdrán Gomáriz, El Gran Libro de la Historia de las Cosas, La Esfera de los Libros, 2009.

[3] B. Schneier, Applied Cryptography, John Wiley & Sons, 1994.

[4] S. Singh, The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography, Doubleday, 1999.

[5] P. Rubens, «Time to Foget your online passwords?» 2014. [En línea]. URL: http://www.bbc.com/future/story/20130703-forget-passwords-take-a-pill. [Último acceso: 06 2014].

[6] T. Hunt, «The only secure Password is the one you can't remember» 2011. [En línea]. URL: http://lifehacker.com/5785420/the-only-secure-password-is-the-one-you-cant-remember. [Último acceso: 06 2014].

[7] P. Ducklin, «Anatomy of a Brute Force Attack» 2013. [En línea]. URL: http://nakedsecurity.sophos.com/2013/08/16/anatomy-of-a-brute-force-attack-how-important-is-password-complexity/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+nakedsecurity+%28Naked+Security+-+Sophos%29. [Último acceso: 06 2014].

[8] R. Smith, Authentication: From Passwords to Public Keys, Addison-Wesley Professional, 2002.

[9] L. L. T. Vrizlynn y Y. Hwei-Ming, «Rainbow Table optimization for Password Recovery» *International Journal on Advances in Software,* vol. 4, pp. 479-488, 2011.

[10] P. N. Patel, J. K. Patel y P. V. Virparia, «A Cryptography Application using Salt Hash Technique» *International Journal of Application or Innovation in Engineering & Management,* vol. 2, nº 6, pp. 236-239, 2013.

[11] M. Tracy, W. Jansen, K. Scarfone y T. Winograd, «Guidelines on Securing Web Servers» *National Institute of Standards and Technology,* Vol. 1 de 2 Special Publication 800-44, 2007.

[12] J. Williams y D. Wichers, OWASP Top 10 - 2013, The Open Web Application Security Project, 2013.

[13] A. Mylonas, A. Kastania y D. Gritzalis, «Delegate the smartphone user? Security awareness in smartphone platforms» *Computers & Security,* vol. 34, pp. 47-66, 2013.

[14] M. Kircher y P. Jain, Pattern-oriented Software Architecture, John Wiley & Sons, 2004.

[15] AETIC, «Infraestructuras de Telefonía Móvil» 2005. [En línea]. URL: http://www.ametic.es/DescargarDocumento.aspx?idd=320. [Último acceso: 06 2014].

[16] E. Martínez, «La evolución de la telefonía móvil» *RED,* Julio 2001.

[17] J. F. Basterretche, «Dispositivos Móviles» 2007. [En línea]. URL: http://repository.lasallista.edu.co/dspace/bitstream/10567/317/3/Dispositivos%20M%C3%B3viles.pdf. [Último acceso: 06 2014].

[18] A. L. Flores Galea, «Evolución de las redes de telefonía móvil» 2009. [En línea]. URL: http://www.antonioflores.es/antonioflores/articulos/consultateleco.pdf. [Último acceso: 06 2014].

[19] A. Baz Alonso, I. Ferreira Artime y M. Álvarez Rodríguez, «Dispositivos Móviles» 2010. [En línea]. URL: http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf. [Último acceso: 06 2014].

[20] F. Serrano Delgado, «BYOD (Tráigase su propio dispositivo, señor empleado público)» *Boletic,* vol. 65, pp. 46-51, 2013.

[21] Citrix, «Best practices to make BYOD simple and secure» 2013. [En línea]. URL: http://www.citrix.com/content/dam/citrix/en_us/documents/oth/byod-best-practices.pdf. [Último acceso: 06 2014].

[22] A. Cavoukian, «BYOD: Is your organization ready?» 2013. [En línea]. URL: http://www.ipc.on.ca/site_documents/pbd-byod.pdf. [Último acceso: 06 2014].

[23] J. Bradley, J. Loucks, J. Macaulay, R. Medcalf y L. Buckalew, «BYOD: Una perspectiva global» 2012. [En línea]. URL: http://www.cisco.com/web/about/ac79/docs/re/byod/BYOD_Horizons-Global_LAS.pdf. [Último acceso: 06 2014].

[24] H. C. Gutiérrez Amaya, «Retos de seguridad para las empresas a partir de BYOD» 2012. [En línea]. URL: http://www.welivesecurity.com/la-es/2012/11/07/nuevo-documento-retos-de-seguridad-para-las-empresas-a-partir-de-byod/. [Último acceso: 06 2014].

[25] M. Souppaya y K. Scarfone, «Guidelines for Managing the Security of Mobile Devices» *National Institute of Standards and Technology,* Vol. 1 de 2 Special Publication 800-124, 2013.

[26] Network Components and Apps. Division at NSA, «Mobile Device Management: A Risk Discussion for IT Decision Makers» 2012. [En línea]. URL: https://www.nsa.gov/ia/_files/factsheets/mdm_decision_makers.pdf. [Último acceso: 06 2014].

[27] G. Sanchidrian, D. Lingenfelter, F. Kasprzykowski y C. Garlati, «Mobile Device Management: Key Components» 2012. [En línea]. URL: https://downloads.cloudsecurityalliance.org/initiatives/mobile/Mobile_Device_Management_Key_Components.pdf. [Último acceso: 06 2014].

[28] Network Intelligence India Pvt. Ltd., «Mobile Device Management - Deployment, Risk Mitigation & Solutions» 2013. [En línea]. URL: http://www.niiconsulting.com/solutions/MDM.pdf. [Último acceso: 06 2014].

[29] M. Báez, Á. Borrego, J. Cordero, L. Cruz, M. González, F. Hernández, D. Palomero, J. Rodríguez de Llera, D. Sanz, M. Saucedo, P. Torralbo y Á. Zapata, Introducción a Android, Madrid: E.M.E. Editorial, 2012.

[30] B. Reed, «A Brief History of Android» 2010. [En línea]. URL: http://dellkv.computerworlduk.com/slideshow/operating-systems/3241860/a-short-history-of-google-android/. [Último acceso: 06 2014].

[31] R. Amadeo, «The History of Android» 2014. [En línea]. URL: http://arstechnica.com/gadgets/2014/06/building-android-a-40000-word-history-of-googles-mobile-os/. [Último acceso: 06 2014].

[32] M. Framingham, «Android and iOS Continue to Dominate the Worldwide Smartphone Market with Android Shipments Just Shy of 800 Million in 2013» 2014. [En línea]. URL: http://www.idc.com/getdoc.jsp?containerId=prUS24676414. [Último acceso: 06 2014].

[33] L. Corrons, «Identificadas Aplicaciones de Google Play que suscriben a SMS premium sin permiso» 2014. [En línea]. URL: http://www.pandasecurity.com/spain/mediacenter/malware/identificadas-aplicaciones-de-google-play-que-suscriben-a-sms-premium-sin-permiso/. [Último acceso: 06 2014].

[34] D. Aryeh, «Google's Verify Apps now features full time app scanning» 2014. [En línea]. URL: http://androidandme.com/2014/04/news/googles-verify-apps-now-features-full-time-app-scanning/. [Último acceso: 06 2014].

[35] Observatorio de la Seguridad de la Información, «Malware y dispositivos móviles» 2012. [En línea]. URL: http://www.inteco.es/Seguridad/Observatorio/Articulos/malwer_moviles. [Último acceso: 06 2014].

[36] T. Eston, OWASP Top 10 Mobile Risks, The Open Web Application Security Project, 2014.

[37] A. Dubey y A. Misra, Android Security: Attacks and Defenses, CRC Press, 2013.

[38] I. Amador, F. Paniagua, D. Suárez y J. M. Sierra, «Mecanismos de autenticación biométrica a través de dispositivos móviles para una ciudad inteligente más segura» 2013. [En línea]. URL: http://innprontaciudad2020.es/index.php/en/documentacion-ficheros-relativos-al-proyecto/white-papers/41-/download. [Último acceso: 06 2014].

[39] B. Wyman, W. Scrivens, P. Hoffman y L. Spitzner, «Autenticación de dos factores» 2012. [En línea]. URL: http://www.securingthehuman.org/newsletters/ouch/issues/OUCH-201211_sp.pdf. [Último acceso: 06 2014].

[40] L. F. Valle Islas, «Coexistencia de redes WLAN & WPAN» 2005. [En línea]. URL: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/valle_i_lf/indice.html. [Último acceso: 06 2014].

[41] J. Penalva, «NFC: ¿qué es y para qué sirve?» 2011. [En línea]. URL: http://www.xataka.com/moviles/nfc-que-es-y-para-que-sirve. [Último acceso: 06 2014].

[42] F. Gallo, «NFC Tags: A technical introduction, applications and products» 2011. [En línea]. URL: http://www.nxp.com/documents/other/R_10014.pdf. [Último acceso: 06 2014].

[43] J. C. Andrés García y S. Okazaki, «El uso de los códigos QR en España» *Distribución y Consumo,* vol. 123, pp. 46-62, 2012.

[44] A. Dmitrienko, C. Liebchen, C. Rossow y A.-R. Sadeghi, «Security Analisis of Mobile Two-Factor Authentication Schemes» *Intel Technology Journal,* vol. 18, nº 4, pp. 138-161, 2014.

Anexos

A continuación se detalla el listado de documentos anexos al documento actual. Dichos documentos presentan información adicional que puede ser de interés o ayudar a comprender esta memoria.

- Anexo 1: Pruebas
- Anexo 2: Manual de Usuario
- Anexo 3: Planificación

SecuBank

Nicolás Mertanen Cuní

Anexo: Pruebas

En este apartado se recoge el resultado de las pruebas realizadas sobre el sistema.

Se realizan dos tipos de pruebas:

- **Pruebas unitarias**: Se verifica el correcto funcionamiento de cada uno de los módulos del sistema.
- **Pruebas de integración**: Se verifica la integración de los distintos componentes del sistema y su funcionamiento de forma conjunta. Se comprueba, a su vez, que se cumple con la funcionalidad especificada anteriormente.

Cada una de las pruebas se compone de los siguientes apartados:

- **Identificador**: Sigue el formato PV-XX, donde P significa Prueba, V puede tomar los valores U para pruebas unitarias, e I para pruebas de integración. Por último, XX indica el valor numérico de la prueba.
- **Descripción**: Descripción de la prueba que se está realizando.
- **Tipo de caja**: Las pruebas se pueden corresponder con dos tipos de caja posibles:
  - **Caja Blanca**: La prueba se centra en analizar el funcionamiento de los métodos.
  - **Caja Negra**: La prueba se centra en que se reciban las salidas esperadas a partir de las entradas introducidas, independientemente del funcionamiento del sistema.
- **Criterios de aceptación**: Valores de respuesta para los cuales se considera que la prueba ha resultado exitosa.
- **Se cumple**: Se utiliza para verificar que se ha recibido el resultado esperado. Puede tomar los valores “Sí” y “No”.
- **Anotaciones**: En caso de que el resultado obtenido no se corresponda con el esperado, se utiliza este apartado para realizar anotaciones sobre anomalías o posibles causas.

PU-01	
Descripción	Login pasa el usuario y la contraseña a LoginCheck por *post*.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los parámetros introducidos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-02	
Descripción	LoginCheck procesa los parámetros recibidos y genera correctamente la sesión para el usuario.
Tipo de caja	Blanca
Criterios de aceptación	Se genera una sesión para el usuario introducido.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-03	
Descripción	Registro pasa a RegCheck los parámetros de registro del usuario por *post*.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los parámetros introducidos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-04	
Descripción	RegCheck recibe los parámetros y, si son correctos, registra al usuario en la BBDD.
Tipo de caja	Blanca
Criterios de aceptación	Se incluyen los datos del usuario en la base de datos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-05	
Descripción	Logout invalida la sesión del usuario.
Tipo de caja	Blanca
Criterios de aceptación	La sesión del usuario queda invalidada.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-06	
Descripción	Se carga correctamente la sesión del usuario al acceder al menú principal.
Tipo de caja	Blanca
Criterios de aceptación	La sesión del usuario está activa.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-07	
Descripción	Movimientos realiza correctamente la consulta sobre la BBDD, mostrando únicamente los movimientos del usuario.
Tipo de caja	Blanca
Criterios de aceptación	Se muestran los movimientos del usuario.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-08	
Descripción	Transferencia envía los campos del formulario a TransCheck por *post*.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los parámetros introducidos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-09	
Descripción	TransConfirma llama a GeneraQR para generar el código QR a mostrar.
Tipo de caja	Blanca
Criterios de aceptación	Se realiza la llamada a las funciones de GeneraQR.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-10	
Descripción	GeneraQR recibe los parámetros necesarios para generar el QR.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben el id de la transferencia, el importe y el username.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-11	
Descripción	Se hace uso de CriptoRSA para cifrar el contenido.
Tipo de caja	Blanca
Criterios de aceptación	El contenido del código QR se encuentra encriptado.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-12	
Descripción	DatosPersonales pasa a DatosCheck la información modificada por *post*.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los parámetros introducidos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-13	
Descripción	DatosCheck procesa los datos y los introduce en la base de datos.
Tipo de caja	Blanca
Criterios de aceptación	Los datos aparecen modificados en la base de datos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-14	
Descripción	AdminUsuarios realiza una consulta sobre toda la tabla de Usuarios de la base de datos.
Tipo de caja	Blanca
Criterios de aceptación	Se recibe la información de todos los usuarios
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-15	
Descripción	AdminDatosUsuario pasa a AdminDatosCheck los parámetros introducidos por *post*.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los datos introducidos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-16	
Descripción	AdminDatosCheck modifica los datos introducidos en la base de datos.
Tipo de caja	Blanca
Criterios de aceptación	Los datos aparecen modificados en la base de datos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-17	
Descripción	AdminTransferencias realiza una consulta sobre toda la tabla Transferencia de la base de datos.
Tipo de caja	Blanca
Criterios de aceptación	Se recibe toda la información de la tabla Transferencias.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-18	
Descripción	AdminNuevoUsuario pasa a AdminNuevoUsuarioCheck los datos introducidos por *post*.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los parámetros introducidos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-19	
Descripción	AdminNuevoUsuarioCheck añade los datos recibidos en la tabla Usuarios de la base de datos.
Tipo de caja	Blanca
Criterios de aceptación	Se crea una nueva entrada con los datos del usuario.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-20	
Descripción	AndroidReg recibe por *post* el nombre de usuario, la contraseña, el identificador del teléfono y la clave pública del usuario.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los parámetros enviados.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-21	
Descripción	AndroidReg ingresa el identificador del teléfono y la clave pública en la base de datos.
Tipo de caja	Blanca
Criterios de aceptación	Se almacena el identificador del terminal en la base de datos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-22	
Descripción	AndroidConn recibe el identificador del teléfono, pasado por el terminal móvil por *post*.
Tipo de caja	Blanca
Criterios de aceptación	Se recibe el identificador del terminal.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-23	
Descripción	AndroidConf recibe los parámetros pasados por el terminal por *post*.
Tipo de caja	Blanca
Criterios de aceptación	Se recibe los parámetros enviados.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-24	
Descripción	AndroidConf descifra el mensaje enviado haciendo uso de la clave privada del servidor.
Tipo de caja	Blanca
Criterios de aceptación	Se obtiene el mensaje en claro.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-25	
Descripción	AndroidConf modifica el registro de la transferencia en caso de haberse confirmado correctamente.
Tipo de caja	Blanca
Criterios de aceptación	La transferencia aparece confirmada.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-26	
Descripción	RegistroActivity envía los parámetros a AndroidReg por *post* a través de la red.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los parámetros enviados.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-27	
Descripción	MainActivity envía el identificador del teléfono a AndroidConn por *post* a través de la red.
Tipo de caja	Blanca
Criterios de aceptación	Se reciben los parámetros enviados.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-28	
Descripción	CapturaActivity decodifica el contenido del código QR, obteniendo el texto en claro.
Tipo de caja	Blanca
Criterios de aceptación	Se obtiene el texto en claro.
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-29	
Descripción	ResultadoActivity desencripta el contenido del QR haciendo uso de la clave privada del usuario.
Tipo de caja	Blanca
Criterios de aceptación	Se obtiene el texto en claro
Se cumple	☒ Sí ☐ No
Anotaciones	

PU-30	
Descripción	ResultadoActivity envía por post la respuesta generada cifrada con la clave pública del servidor.
Tipo de caja	Blanca
Criterios de aceptación	Se recibe la respuesta en el servidor.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-01	
Descripción	Cuando el usuario se registra en el sistema, puede hacer login en el mismo.
Tipo de caja	Negra
Criterios de aceptación	Se hace login con los datos introducidos en el registro.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-02	
Descripción	Si se introducen datos erróneos en el formulario de registro, se avisa con el mensaje de error pertienente.
Tipo de caja	Negra
Criterios de aceptación	Se recibe un mensaje de error.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-03	
Descripción	Desde el menú se tiene acceso a toda la funcionalidad del sistema.
Tipo de caja	Negra
Criterios de aceptación	Se puede acceder a movimientos, transferencias y datos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-04	
Descripción	En movimientos se pueden consultar todos los movimientos realizados sobre la cuenta del usuario.
Tipo de caja	Negra
Criterios de aceptación	Se muestran tanto ingresos como pagos.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-05	
Descripción	Cuando se introducen los datos para realizar una transferencia, se genera un código QR que escanear con el móvil.
Tipo de caja	Negra
Criterios de aceptación	Aparece un QR en la pantalla de confirmación.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-06	
Descripción	Si los datos introducidos en el formulario de la transferencia no son correctos, se genera un mensaje de error.
Tipo de caja	Negra
Criterios de aceptación	Se recibe un mensaje de error.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-07	
Descripción	Cuando se presiona sobre Información Personal, se muestran los datos personales del usuario.
Tipo de caja	Negra
Criterios de aceptación	Se ven los datos del usuario.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-08	
Descripción	Cuando se modifican los datos del usuario, dichos cambios se reflejan en la base de datos.
Tipo de caja	Negra
Criterios de aceptación	Al acceder de nuevo a Información, los datos están modificados.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-09	
Descripción	Si los datos contienen errores, se informa al usuario con un mensaje de error.
Tipo de caja	Negra
Criterios de aceptación	Se recibe un mensaje de error.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-10	
Descripción	Al hacer logout, el usuario se desconecta del sistema.
Tipo de caja	Negra
Criterios de aceptación	Se recibe un mensaje de desconexión.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-11	
Descripción	El administrador puede consultar la lista completa de usuarios.
Tipo de caja	Negra
Criterios de aceptación	Se obtiene una tabla con todos los usuarios del sistema.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-12	
Descripción	El administrador puede consultar la información personal del usuario.
Tipo de caja	Negra
Criterios de aceptación	Se muestra un formulario con toda la información del usuario.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-13	
Descripción	El administrador puede modificar la información de un usuario.
Tipo de caja	Negra
Criterios de aceptación	La información aparece modificada cuando se vuelve a consultar.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-14	
Descripción	El administrador puede consultar el listado completo de transferencias realizadas.
Tipo de caja	Negra
Criterios de aceptación	Se muestra una tabla con todas las transferencias del sistema.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-15	
Descripción	El administrador puede dar de alta a nuevos usuarios.
Tipo de caja	Negra
Criterios de aceptación	Se puede acceder al sistema desde el usuario creado por el admin.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-16	
Descripción	Solamente se solicita la confirmación del registro desde la aplicación móvil la primera vez que se arranca ésta.
Tipo de caja	Negra
Criterios de aceptación	La siguiente vez accede al menú principal.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-17	
Descripción	Si no se dispone de conexión, la aplicación móvil no se puede utilizar.
Tipo de caja	Negra
Criterios de aceptación	Sin conexión no se accede a la aplicación.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-19	
Descripción	Cuando se presiona sobre el botón de captura de la aplicación móvil, aparece la cámara.
Tipo de caja	Negra
Criterios de aceptación	Se muestra la vista de cámara del dispositivo.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-20	
Descripción	Si se escanea un QR con el dispositivo, se intenta descifrar de forma automática.
Tipo de caja	Negra
Criterios de aceptación	Al capturar un QR, la pantalla cambia a “procesando” al instante.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-21	
Descripción	Si el código se corresponde con una transferencia de la aplicación generada por el usuario, ésta queda confirmada.
Tipo de caja	Negra
Criterios de aceptación	Se recibe un mensaje “Transferencia confirmada”.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-22	
Descripción	Si el código no pertenece al usuario, la transferencia no se confirma.
Tipo de caja	Negra
Criterios de aceptación	Se recibe un mensaje de error “Transferencia rechazada”.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-23	
Descripción	Las contraseñas almacenadas en la base de datos se almacenan haciendo uso de la función Hash.
Tipo de caja	Negra
Criterios de aceptación	Se visualizan los Hash al consultar la BBDD.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-24	
Descripción	La información del servidor se cifra mediante un algoritmo AES para su guardado.
Tipo de caja	Negra
Criterios de aceptación	La información se encuentra cifrada.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-25	
Descripción	La base de datos se encuentra protegida por contraseña.
Tipo de caja	Negra
Criterios de aceptación	Al acceder se solicita la contraseña.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-26	
Descripción	La clave privada del usuario se guarda cifrada con un algoritmo AES.
Tipo de caja	Negra
Criterios de aceptación	La clave aparece cifrada.
Se cumple	☒ Sí ☐ No
Anotaciones	

PI-27	
Descripción	Las dos aplicaciones que conforman el sistema guardan una relación de aspecto consecuente.
Tipo de caja	Negra
Criterios de aceptación	El aspecto de ambas aplicaciones es similar.
Se cumple	☒ Sí ☐ No
Anotaciones	

Matriz de trazabilidad: Pruebas – Requisitos de software

A continuación se muestra una matriz que permite verificar que cada uno de los requisitos de software se ha verificado con al menos una de las pruebas realizadas.

	RSRF01	RSRF02	RSRF03	RSRF04	RSRF05	RSRF06	RSRF07	RSRF08	RSRF09	RSRF10	RSRF11	RSRF12	RSRF13	RSRN01	RSRN02	RSRN03	RSRN04	RSRN05	RSRN06	RSRN07	RSRN08
U 0 1		X																			
U 0 2		X																			
U 0 3	X																				
U 0 4	X																				
U 0 5		X							X												
U 0 6																					
U 0 7			X																		
U 0 8						X															
U 0 9							X														
U 1 0							X														
U 1 1							X														
U 1 2				X	X																
U 1 3					X																
U 1 4										X											
U 1 5											X										
U 1 6											X										
U 1 7												X									
U 1 8													X								
U 1 9													X								
U 2 0																					
U 2 1																					
U 2 2														X							
U 2 3																					
U 2 4																					
U 2 5																					
U 2 6																					
U 2 7														X							

	RSRF01	RSRF02	RSRF03	RSRF04	RSRF05	RSRF06	RSRF07	RSRF08	RSRF09	RSRF10	RSRF11	RSRF12	RSRF13	RSRN01	RSRN02	RSRN03	RSRN04	RSRN05	RSRN06	RSRN07	RSRN08
U28																					
U29																					
U30																					
I01	X	X																			
I02															X			X	X	X	
I03								X													
I04			X																		
I05						X	X														
I06																					
I07				X																	
I08					X																
I09																					
I10									X												
I11																					
I12										X											
I13											X										
I14												X									
I15													X								
I16																					
I17																					
I18																					
I19																					
I20																					
I21																					
I22																					
I23																X					
I24																	X				
I25																					
I26																					
I27																					X

SecuBank

Nicolás Mertanen Cuní

Anexo: Manual de Usuario

Introducción

SecuBank es un sistema bancario donde prima la seguridad. El objetivo es garantizar tanto la privacidad de los datos y la información de los usuarios como las transacciones y el movimiento de fondos realizado por los mismos.

Instalación y uso

Para el acceso a la plataforma web basta con acceder al dominio en que se encuentra alojada. Adicionalmente, para poder hacer uso de todas las funciones que ofrece la plataforma, es necesario instalar una aplicación en el Smartphone. El proceso de instalación es el siguiente:

Se debe descargar el archivo instalable “SecuBank.apk”. Una vez descargado dicho archivo en el dispositivo, se abre para proceder a su instalación.

El instalador muestra los permisos que necesita la aplicación para funcionar correctamente, y solicita la confirmación del usuario para proceder con su instalación. Los requisitos necesarios para su correcto funcionamiento son los siguientes:

- INTERNET: Acceso a Internet sin límites.
- READ_PHONE_STATE: Leer ID y estado del teléfono.
- ACCESS_WIFI_STATE: Ver estado de conexión, ver estado Wi-Fi.
- CAMERA: Realizar fotografías.

Para dar su confirmación se debe presionar sobre Instalar, con lo que se mostrará una barra de progreso donde se puede observar el proceso de instalación de la aplicación. Se debe disponer de, al menos, 7,75MB disponibles de espacio libre en el teléfono.

Una vez finalizada la instalación, se ofrecen las opciones Abrir y Hecho. Abrir arranca la aplicación por primera vez, mientras que Hecho cierra el instalador y vuelve al menú anterior.

El icono de la aplicación es el siguiente:

SecuBank

3.1 Primer acceso y registro

El acceso al sistema presenta la pantalla de login, que ofrece la posibilidad de registrarse a nuevos usuarios. El proceso de registro es necesario para poder acceder al sistema y hacer uso de toda la funcionalidad que ofrece.

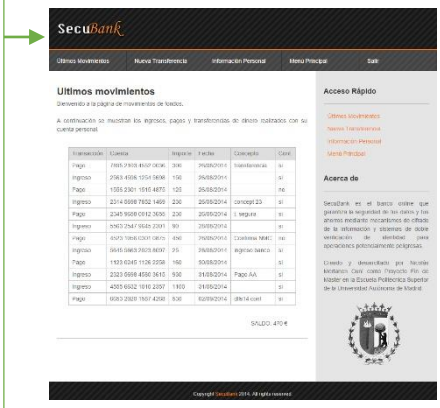
El proceso de registro se compone de dos pasos. En primer lugar se debe acceder a la página de registro y rellenar los datos personales y de acceso requeridos:

En segundo lugar se debe instalar la aplicación móvil para confirmar las operaciones solicitadas e ingresar con los datos de acceso introducidos en el proceso de registro:

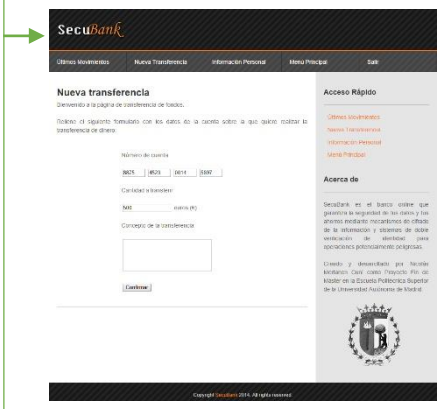
Una vez completado, el usuario se encuentra dado de alta en el sistema.

3.2 Operaciones sobre el sistema

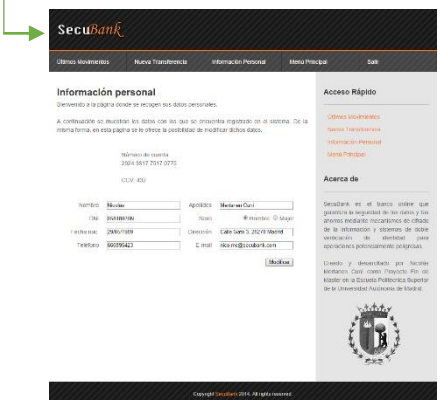
Una vez registrado, el usuario puede acceder al sistema con sus datos de acceso. Una vez ha ingresado, se encuentra en el Menú principal de la aplicación, desde donde tiene acceso a la siguiente funcionalidad:



Últimos movimientos: Listado de movimientos realizados sobre la cuenta del usuario. Se detalla tanto el estado de las transferencias realizadas por el usuario (confirmada o no confirmada), como los ingresos recibidos sobre su cuenta.



Nueva transferencia: Formulario que permite la realización de una transferencia sobre la cuenta de otro usuario. Dicho formulario se compone de un número de cuenta, una cantidad de dinero a transferir y un concepto de la transferencia (opcional).

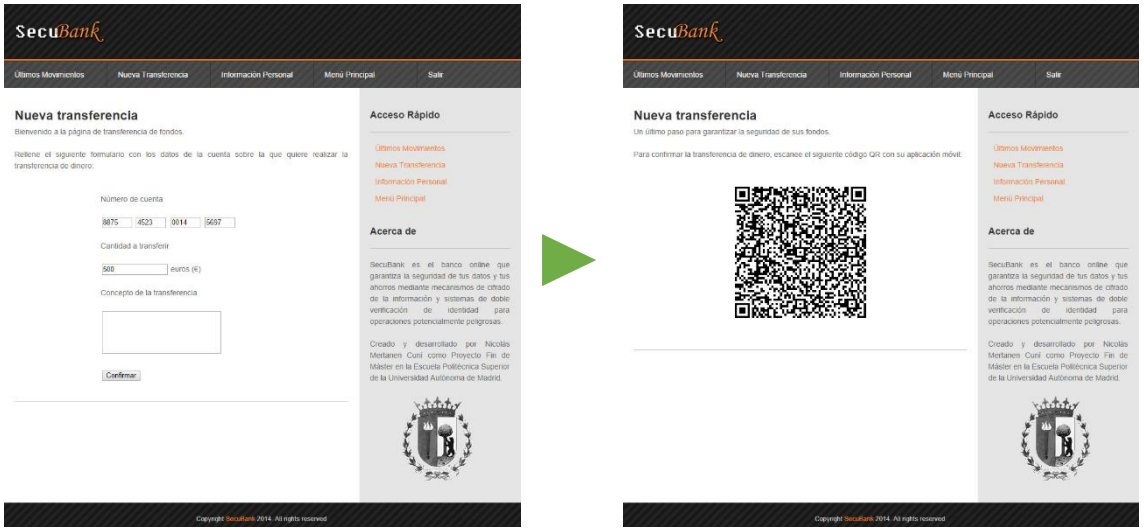


Información personal: Contiene la información con que se ha registrado el usuario en el sistema. Adicionalmente presenta el número de cuenta y el código CCV y permite al usuario modificar la información con que se dio de alta.

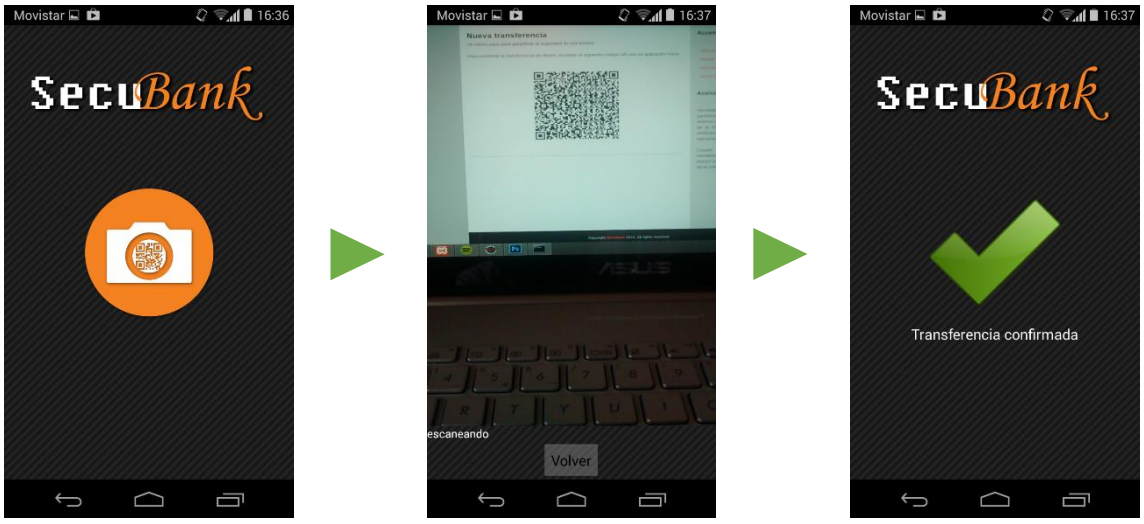
3.2.1 Transferencia

El procedimiento necesario para realizar una transferencia implica la utilización conjunta de la aplicación web y la aplicación móvil.

En el portal web se solicita la transferencia de dinero a través de “Nueva transferencia”. Una vez rellenados los datos solicitados, se presiona sobre confirmar, con lo que aparece un código QR que contiene la información relativa a la transferencia para su confirmación.



Dicho código QR tiene que ser escaneado con la aplicación móvil para confirmar la transferencia. Una vez se recibe el mensaje de confirmación, la transferencia ha sido aceptada y el dinero se ha transferido de manera satisfactoria.



SecuBank

Nicolás Mertanen Cuní

Anexo: Planificación

La planificación del proyecto recoge el listado de roles desempeñado, el presupuesto total del proyecto y el análisis de tiempos.

Listado de roles y perfiles

El listado de funciones necesarias para el correcto desarrollo del proyecto es el siguiente:

- **Jefe de proyecto**: Es el encargado de dirigir el desarrollo del proyecto, garantizando que se cumplen los requisitos necesarios para su correcta implementación. Debe seguir de cerca todo el desarrollo, desde su estudio inicial hasta su puesta en funcionamiento.
- **Analista**: Es el encargado de comunicarse con el cliente para conocer cuáles son sus exigencias con respecto al sistema a desarrollar. Dichas exigencias serán plasmadas en forma de requisitos de usuario.
- **Diseñador**: Es el encargado de plantear una arquitectura que cumpla con los requisitos solicitados por el cliente, permitiendo la posterior implementación del sistema.
- **Programador**: Es el encargado de llevar a cabo la implementación del sistema a partir de la arquitectura detallada anteriormente. Una vez realizada la implementación, se llevará a cabo una serie de pruebas para garantizar que el sistema funciona de forma correcta y se cumplen todos los requisitos satisfactoriamente.

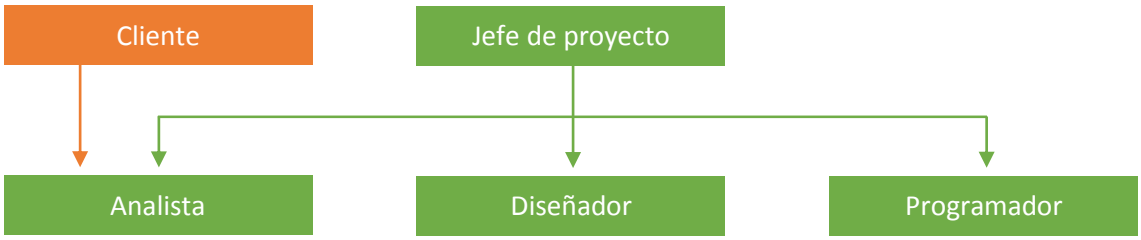
A su vez se dispone de un agente externo al equipo de desarrollo que también influye en el progreso del sistema:

- **Cliente**: Es la persona o entidad que contrata los servicios del equipo de desarrollo y solicita la implementación del sistema que se va a desarrollar.

Como el proyecto actual es desarrollado por una sola persona, ésta tomará cada uno de los roles detallados anteriormente según el momento en que se encuentre del desarrollo del sistema.

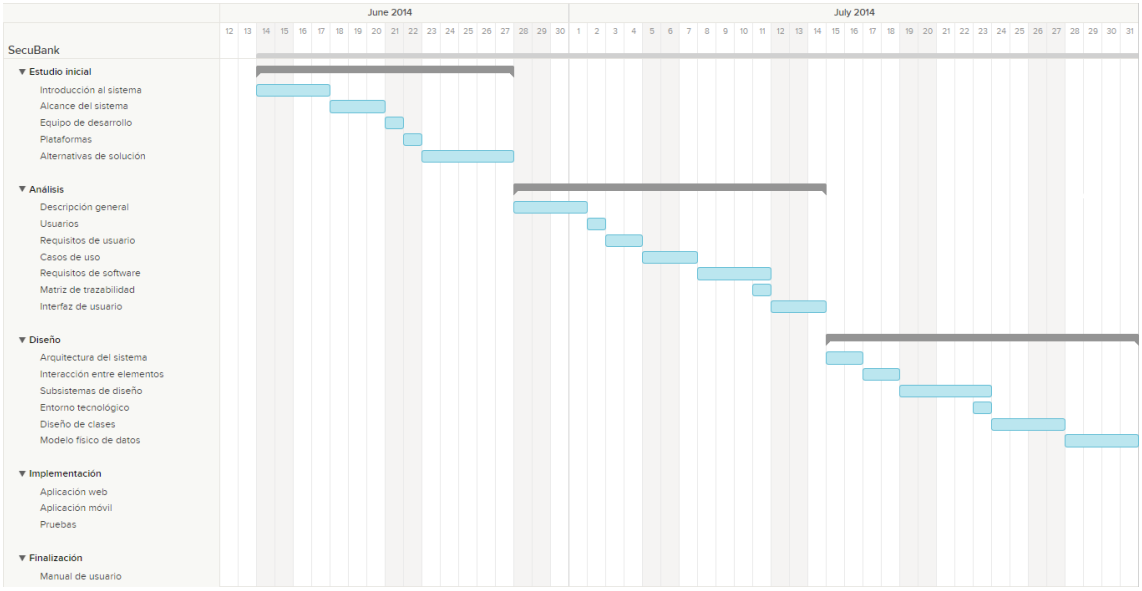
Organigrama

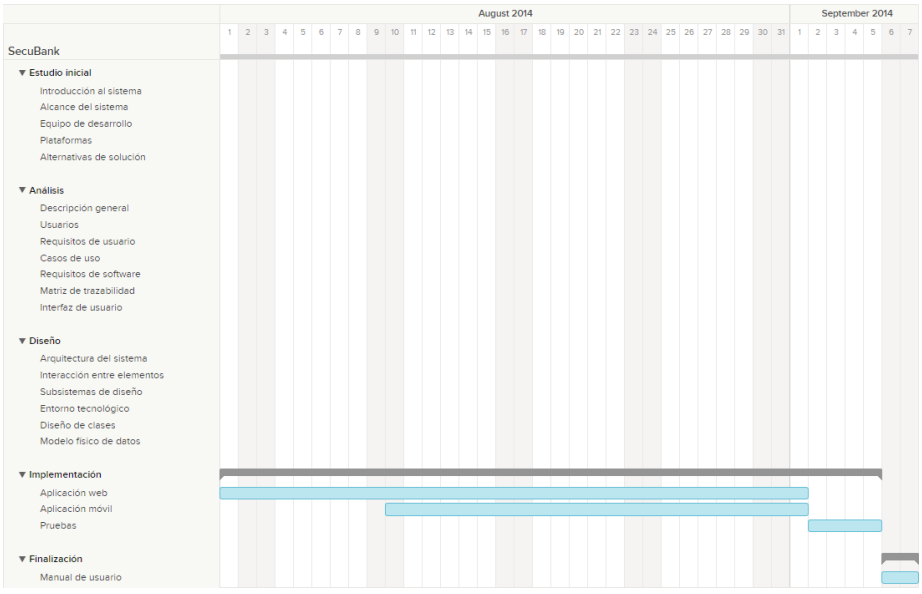
El siguiente esquema determina la relación entre los distintos roles definidos anteriormente:



Planificación de tiempos

El cálculo de tiempos estimado se representa mediante un Diagrama de Gantt, que permite visualizar de forma sencilla el tiempo empleado en la realización de cada sección del proyecto.





El cálculo del coste total del proyecto se desglosa de la siguiente forma:

4.1 Recursos Humanos

En la siguiente tabla se representa el sueldo bruto para cada uno de los roles que intervienen en el proyecto y el total de horas de trabajo estimadas, permitiendo el cálculo del coste total en recursos humanos del proyecto.

Recursos Humanos			
Rol	Sueldo/hora	Horas de trabajo	Total (€)
Jefe de proyecto	22,00	180	3.960,00
Analista	18,00	100	1.800,00
Diseñador	14,00	110	1.540,00
Programador	10,00	210	2.100,00
			9.400,00

4.2 Amortizaciones

A continuación se realiza un cálculo del periodo y coste de amortización de equipos y licencias, ofreciendo el coste final imputable sobre el proyecto.

Amortizaciones			
Equipo	Coste	Periodo amortización	Total (€)
Equipo de desarrollo	1200,00	5 años	50,00
Terminal de pruebas	180,00	2 años	18,75
Microsoft Office '13	500,00	5 años	21,00
Adobe Photoshop cs6	270,00	5 años	11.25
			101,00

4.3 Costes indirectos

Por último, se detalla el cálculo de gastos no contemplados en las tablas anteriores.

Costes indirectos		
Concepto	Coste	Total (€)
Luz y agua	150,00	150,00
Material fungible	25,00	25,00
Teléfono e internet	45,00	45,00
		220,00

4.4 Cálculo total de costes

Una vez desglosado cada uno de los costes, se realiza un resumen de costes total del proyecto.

Cálculo total de costes	
Concepto	Total (€)
Recursos humanos	9.400,00
Amortizaciones	101,00
Costes indirectos	220,00
	9.721,00
I.V.A. (21%)	2.041,41
	11.762,41

El coste total del proyecto es de 11.762,41 €.